1 ПОТОКИ В СЕТЯХ

Определение 1. Сеть $N = (V, E, s, t, c)$ *- это связный ориентированный граф без петель* $G = (V, E)$, *у которого*

1) имеется одна вершина s, *в которую не входят дуги (источник);*
2) имеется одна вершина t, *из которой не исходят дуги (сток);*
3) каждой дуге $e = (u, v) \in E$ *сопоставлена ее пропускная способность* $c(e) \geq 0$.

Определение 2. Поток f в сети N *- это функци* $f : E \to R$ *такая, что*

1) $0 \leq f(e) \leq c(e)$ *для каждой дуги* $e \in E$;
2) для каждой вершины $v \in V \setminus \{s, t\}$
$\sum\{f(e)|\ e \text{ входит в } v\} = \sum\{f(e)|\ e \text{ выходит из } v\}$.

Величина потока f *в сети* N $val(f) = \sum_{u \in V} f(s, u)$.

Следствие. Величина потока $val(f) = \sum_{u \in V} f(u, t)$. (Показать!)

Определение 3. Разрез A в сети N *- подмножество вершин* A *такое, что источник* $s \in A$, *а сток* $t \in V \setminus A$. *Разрез (и вообще любое подмножество вершин сети)* A *однозначно задает множество дуг* $P(A) = \{(u, v) \mid\ u \in A, v \in V \setminus A\}$, *ведущих из* A *"наружу".*

Пропускная способность разреза A *:*

$$c(A, V \setminus A) = \sum_{(u,v) \in P(A)} c(u, v)$$

Разрез с минимальной пропускной способностью называетс минимальным.

Поток f через разрез:

$$f(A, V \setminus A) = \sum_{(u,v) \in P(A)} f(u, v)\ .$$

Лемма 1. *Для произвольного разреза* A *и любого потока* f

$$val(f) = f(A, V \setminus A) - f(V \setminus A, A).$$

Доказательство. Из определения потока следует, что

$$\sum_{v \in V} f(u, v) - \sum_{v \in V} f(v, u) = \begin{cases} val(f) & \text{если u = s} \\ 0 & \text{если } u \in A \setminus \{s\} \end{cases}$$

Просуммировав эти равенства по всем $u \in A$, получим

$$\sum_{u \in A} \sum_{v \in V} f(u, v) - \sum_{u \in A} \sum_{v \in V} f(v, u) = val(f).$$

Для любой пары вершин $u \in A, v \in A$ поток $f(u, v)$ учитывается в первой сумме в левой части с плюсом, а во второй сумме с минусом. Сократив все такие слагаемые, получим

$$\sum_{u \in A} \sum_{v \in V \setminus A} f(u, v) - \sum_{u \in A} \sum_{v \in V \setminus A} f(v, u) = val(f)$$

или $f(A, V \setminus A) - f(V \setminus A, A) = val(f)$.

Теорема 1 ((Форд, Фалкерсон)).
(1) Величина любого потока из s *в* t *не превосходит пропускной способности минимального разреза. (2) Существует поток, достигающий этого значения (максимальный поток).*

Доказательство. (1) Пусть A — минимальный разрез. По лемме 1 для любого потока f имеем $val(f) = f(A, V \setminus A) - f(V \setminus A, A) \leq f(A, V \setminus A) = \Sigma_{e \in P(A)} f(e) \leq \Sigma_{e \in P(A)} c(e) = c(A, V \setminus A)$.

Определение 4. *Пусть N - сеть, f - поток для нее.* Увеличивающий путь (увеличивающая цепь) *– это любой путь P из s в t в неориентированном графе, полученном из G игнорированием направлений дуг, в котором:*

1) для любой дуги (u,v), проходимой в P в прямом направлени (прямой дуги), $f(u,v) < c(u,v)$;

2) для любой дуги (v,u), проходимой в P в обратном направлении (обратной дуги), $f(v,u) > 0$.

Теорема 2. *Следующие условия эквивалентны:*

1) поток f из s в t максимальный;

2) не существует увеличивающего пути для f;

3) для некоторого разреза A $val(f) = c(A, V \setminus A)$.

Доказательство. (1) $\Longrightarrow$ (2) следует из определения увеличивающего пути. Действительно, предположим, что в сети есть увеличивающий путь $p = (s = v_0 \stackrel{e_1}{\to} v_1 \stackrel{e_2}{\to} \dots, v_{l-1} \stackrel{e_l}{\to} v_l = t)$. Положим $\Delta(e_i) = c(e_i) - f(e_i)$, если дуга e_i прямая, и $\Delta(e_i) = f(e_i)$, если e_i — обратная дуга. Пусть $\delta = \min\{\Delta(e_i) | 1 \leq i \leq l\}$. Тогда $\delta > 0$. Определим новый поток f', который совпадает с f всюду кроме p, а на дугах из p положим $f'(e_i) = f(e_i) + \delta$, если дуга e_i прямая, и $f'(e_i) = f(e_i) - \delta$, если дуга e_i обратная. Нетрудно проверить, что f' является потоком и что $val(f') = val(f) + \delta$.

(2) $\Longrightarrow$ (3). Положим $A = \{v \in V |$существует увеличивающая цепь $s = v_0 \stackrel{e_1}{\to} v_1 \stackrel{e_2}{\to} \dots, v_{k-1} \stackrel{e_k}{\to} v_k = v\}$. Тогда A — разрез (почему?). Для каждой дуги $e \in P(A)$ имеет место $f(e) = c(e)$ (иначе можно было бы продолжить увеличивающую цепь). По той же причине для каждой дуги $e \in P(V \setminus A)$ $f(e) = 0$. Тогда по лемме 1 получаем, что $val(f) = f(A, V \setminus A) - f(V \setminus A, A) = c(A, V \setminus A)$.

(3) $\Longrightarrow$ (1) следует по теореме 1.

АЛГОРИТМ ФОРДА И ФАЛКЕРСОНА (Ф-Ф)

Вход: сеть $N = (V, E, s, t, c)$.

Выход: максимальный поток f в N.

Структуры данных:

$L1(v)$ - предшественник v в ув. пути, $L2(v)$ - величина дополнительного потока из s в v, СПИСОК - очередь вершин.

```
BEGIN
        FOR ALL  e ∈ E  DO  f(e) := 0  END_DO;
ЕЩЕ_РАЗ: FOR ALL v ∈ V \ {s}  DO  L1(v) := 0;  L2(v) := 0  END_DO;
        L1(s) := s;  L2(s) := ∞;
        СПИСОК := {s};
        WHILE  СПИСОК  ≠ ∅  DO
               BEGIN выбрать вершину v из СПИСОК;
                   удалить v из  СПИСОК;
                   ПРОСМОТРЕТЬ(v);
                   IF  L1(t) ≠ 0  %  t помечена, есть увел. путь
                   THEN BEGIN
(*)                       увеличить поток f вдоль увеличивающего пути;
                          GO TO ЕЩЕ_РАЗ
        END END END
            PROCEDURE ПРОСМОТРЕТЬ(v)
BEGIN
1) для каждой  u,  такой что (v,u) ∈ E,  L1(u) = 0 и f(v.u) < c(v,u),
        цикл   % пометить u:
               L1(u) := v;  L2(u) := min{L2(v), c(v,u) − f(v,u)};
               СПИСОК := СПИСОК ∪{u};
        все_цикл;
2) для каждой u, такой что  (u,v) ∈ E,  L1(u) = 0 и f(u,v) > 0,
```

цикл % пометить u :

положить $L1(u) := -v$; $L2(u) := min\{L2(v), f(v,u)\}$;

СПИСОК := СПИСОК $\cup\{u\}$;

все_цикл;

END

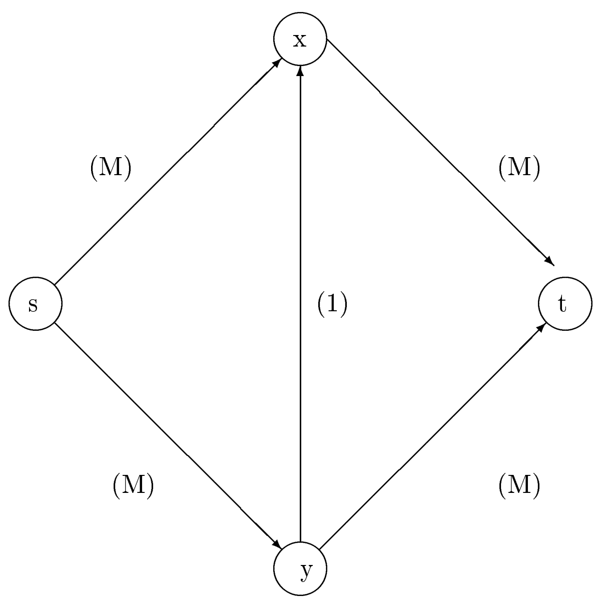
Теорема 3. *Если алгоритм Ф-Ф останавливается, то полученный поток максимален.*

Доказательство. Действительно, в момент остановки алгоритма Ф-Ф СПИСОК пуст и $L1(t) = 0$. Положим $W = \{v \mid L1(v) \neq 0\}$. Тогда W – разрез. Для любой дуги $(v,u) \in P(W)$ $f((u,v) = c((u,v))$ (иначе при вызове ПРОСМОТРЕТЬ(v) на этапе 1) для u были бы выполнены все условия и эта вершина была бы "помечена": $L1(u) := v$. Для любой же обратной дуги $(u,v) \in P(V \setminus W)$ $f((u,v) = 0$ (иначе при вызове ПРОСМОТРЕТЬ(v) на этапе 2) для u были бы выполнены все условия и эта вершина была бы "помечена": $L1(u) := -v$). Но тогда по лемме 1 имеем

$val(f) = f(W, V \setminus W) - f(V \setminus W, W) = c(W, V \setminus W)$

и по теореме 2 f – максимальный поток.

Следующий пример показывает, что сложность алгоритма Ф-Ф может зависеть от значений пропускных способностей дуг. Рассмотрим следующую сеть:



Предположим, что алгоритм Ф-Ф, начав с нулевого потока, поочередно находит и использует для увеличения потока увеличивающие пути $p_1 = s \to y \to x \to t$ и $p_2 = s \to x \leftarrow y \to t$. Тогда на каждом этапе алгоритма поток увеличивается на 1 и максимальный поток $2M$ достигается за $2M$ этапов.

ЗАМЕЧАНИЕ. Имеются примеры при иррациональных значениях $c(v,u)$, когда алгоритм не останавливается и сходится не к мак- симальному значению.

Определение 5. *Пусть* $N = (V, E, s, t, c)$ *– сеть,* f *– поток в* N*.* Вспомогательная сеть $AN(f) = (V(f), E(f), s, t, ac)$ *– это такая сеть, вершины которой разбиты на слои* $V_0 = \{s\}, V_1, ...,$ $V_l = \{t\}$*, так что вершины слоя* V_i *находятся на расстоянии* i *от* s*, дуги соединяют вершины соседних слоев* V_i *и* V_{i+1} *и при этом*

1) если $(u, v) \in E,\ u \in V_i,\ v \in V_{i+1}$ *и* $f(u, v) < c(u, v)$*, то* $ac(u, v) = c(u, v) - f(u, v)$*;*

2) если $(v, u) \in E,\ u \in V_i,\ v \in V_{i+1}$ *и* $f(v, u) > 0$*, то* $ac(u, v) = f(v, u)$*;*

3) если $(u, v) \in E,\ (v, u) \in E,\ u \in V_i,\ v \in V_{i+1}$*, то* $ac(u, v) = c(u, v) - f(u, v) + f(v, u)$*.*

Алгоритм ПОВС построения вспомогательной сети основан на поиске в ширину. Пусть $L(v) = \{u \mid (v, u) \in E\}$ – список "сыновей"v, $M(v) = \{u \mid (u, v) \in E\}$ – список "отцов"v в N, а $AL(v)$ и $AM(v)$ – аналогичные списки для $AN(f)$, AV – вершины $AN(f)$, ДЛИНА(v) - расстояние от s до v.

```
    PROCEDURE ПОВС
BEGIN                          % Инициализация:
1         FOR ALL  u ∈ V  DO
2               BEGIN  ДЛИНА(u) := ∞;
3                     AM(u) := ∅; AL(u) := ∅;
4                     FOR ALL  v ∈ V  DO {ac(u,v) := 0;}
5                     FOR ALL  v ∈ V  DO  {af(u,v) := 0;}
                END;
6         ОЧЕРЕДЬ  := {s};  AV := ∅;  ДЛИНА(s) := 0;
                   % поиск в ширину от s:
7         WHILE ОЧЕРЕДЬ  ≠ ∅  DO
8               BEGIN  u := НАЧАЛО(ОЧЕРЕДЬ); УДАЛИТЬ(ОЧЕРЕДЬ,u);
                          AV := AV ∪ {u};
9                     FOR ALL  v ∈ L(u)  DO             % поиск прямых дуг
10                       IF (ДЛИНА(u) <  ДЛИНА(v) ≤ ДЛИНА(t))AND
11                             f(u,v) < c(u,v))  THEN
12                       BEGIN IF  ДЛИНА(v) = ∞                  % v - новая
13                          THEN  ДОБАВИТЬ(ОЧЕРЕДЬ, v);
14                          ДЛИНА(v) := ДЛИНА(u) + 1;
                   % добавить (u,v) к сети AN:
15                          AL(u) := AL(u) ∪ {v};  AM(v) := AM(v) ∪ {u};
16                          ac(u,v) := c(u,v) − f(u,v)
                         END;
17                    FOR ALL  v ∈ M(u)  DO             % поиск обратных дуг
18                       IF  (ДЛИНА(u) < ДЛИНА(v) ≤ ДЛИНА(t))  AND
19                             f(v,u) > 0)  THEN
20                       BEGIN IF  ДЛИНА(v) = ∞ THEN                 % v - новая
21                          ДОБАВИТЬ(ОЧЕРЕДЬ,v);
22                          ДЛИНА(v) := ДЛИНА(u) + 1;
23                          IF  ac(u,v) = 0  THEN % добавить (u,v) к сети AN
24                             BEGIN  AL(u) := AL(u) ∪ {v};  AM(v) := AM(v) ∪ {u}
                                END;
25                    ac(u,v) := ac(u,v) + f(v,u)
             END      END;
26              удалить из  AN(F)  все вершины, из которых
                 нет пути в  t  и инцидентные им дуги
END
```

Теорема 4. *а) Если в результате работы алгоритма ПОВС ДЛИНА$(t) = \infty$, то в сети N нет увеличивающего пути от s к t и, следовательно, поток f максимальный.*

б) Если ДЛИНА$(t) = l \neq \infty$, то кратчайший увеличивающий путь из s в t в сети N имеет длину l. Более того, пути из s в t в сети AN взаимно однозначно соответствуют увеличивающим путям длины l в сети N.

в) Если $maxf$ – величина максимального потока в N, то величина максимального потока в $AN(f)$ равна $maxf - val(f)$.

г) Время работы процедуры ПОВС – $O(|E|) \leq O(n^2)$, где $n = |V|$.

Определение 6. *Поток g в сети AN называется* тупиковым, *если для него не существует увеличивающего пути без обратных дуг (*прямого *увеличивающего пути).*

Определение 7. Потенциал $cnc(v)$ *вершины v в сети N - это максимальное количество потока, которое можно протолкнуть через v, т.е. $cnc(v) = min(\sum_{u\in V} c(u,v) \sum_{u\in V} c(v,u))$ для $v \in V \setminus \{s,t\}$, $cnc(s) = \sum_{u\in V} c(s,u)$, $cnc(t) = \sum_{u\in V} c(u,t)$.*

АЛГОРИТМ МАХП ПОСТРОЕНИЯ МАКСИМАЛЬНОГО ПОТОКА

Вход: сеть $N = (V, E, s, t, c)$.

Выход: максимальный поток f в N.

```
BEGIN
  FOR ALL v ∈ V DO f(u,v) := 0; готово := "нет";
  WHILE готово="нет" DO
     BEGIN % новый этап
        ПОВС;              % построить вспомогательную сеть AN(f)
             % Построение тупикового потока:
        FOR ALL (u,v) ∈ E(f) DO
                   g(u,v) := 0;          %инициал. тупикового потока
        IF t недостижима из s в AN(f)
          THEN готово := "да"
        ELSE WHILE cnc(s) > 0 AND cnc(T) > 0 DO
                   BEGIN
                   найти v с минимальным потенциалом cnc(v);   (> 0)
                    ПРОТОЛКНУТЬ(v, cnc(v));
                    ПРОТЯНУТЬ(v, cnc(v))
                    WHILE есть v ∈ V(f) \ {s,t}, для которой cnc(v) = 0
                          DO удалить v и все инцидентные ей дуги из AN(f);
                    END
             % Добавление тупикового потока:
         FOR ALL (u,v) ∈ E DO
                        IF (u,v) ∈ E(f)    % прямая дуга
                        THEN f(u,v) := f(u,v) + g(u,v)
                        ELSE IF (v,u) ∈ E(f)   % обратная дуга
                              THEN f(u,v) := f(u,v) − g(u,v) END_DO
    END    END
```

PROCEDURE ПРОТОЛКНУТЬ(y,h)

увеличивает поток g на h единиц, проталкиваемых от y к t

```
BEGIN
  создать очередь Q; ДОБАВИТЬ(Q, y);
  FOR ALL u ∈ V(f) \ {y}
DO треб[u] := 0;   % размер выталкиваемого из u и y потока
```

```
3         треб[y] := h;
4        WHILE  Q ≠ ∅  DO
5              BEGIN  v :=  НАЧАЛО(Q);  УДАЛИТЬ(Q, v);
6              FOR ALL  v' ∈ AL(v)  and UNTIL  треб[v] = 0  DO
7                     BEGIN
8                     m := min(ac(v, v'), треб[v]);
9                     ac(v.v') := ac(v, v') − m;
10                  IF  ac(v, v') = 0  THEN  E(f) := E(f) \ {(v, v')};
11                   треб[v] := треб[v] − m;
12                   треб[v'] := треб[v'] + m;
13                    ДОБАВИТЬ(Q, v');
14                   g(v, v') := g(v, v') + m
15 END          END          END
```

PROCEDURE ПРОТЯНУТЬ(y,h)
увеличивает поток g на h единиц, протягиваемых от y к s
Алгоритм аналогичен процедуре ПРОТОЛКНУТЬ.

Лемма 2. *Дуга e сети $AN(f)$ удаляется из $E(f)$ на некотором шаге только в том случае, если в $AN(f)$ нет прямого увеличивающего пути относительно потока g, проходящего через e.*

Доказательство. Так как для любой дуги e в начале этапа $g(e) = 0$, а затем увеличение потока и уменьшение пропускной способности e происходят одновременно и на одну и ту же величину (стр. 9 и 14 процедуры ПРОТОЛКНУТЬ), то в каждый момент (после стр. 14) $g(e)+ac(e) = ac'(e)$, где $ac'(e)$ – начальна пропускная способность дуги e. Поэтому, если эта дуга удаляется в стр. 10 процедуры ПРОТОЛКНУТЬ, то $g(e) = ac'(e)$ и e не может входить в прямой увеличивающий путь в $AN(f)$. Если же $e = (u, v)$ удаляется в стр. 11 МАХП, то либо $cnc(u) = 0$, либо $cnc(v) = 0$. В первом случае для всякой входящей в u дуги e_1 $g(e_1) = ac'(e_1)$, а во втором дл всякой выходящей из v дуги e_2 $g(e_2) = ac'(e2)$. В обоих случаях это означает, что нет прямого увеличивающего пути через e.

Лемма 3. *В конце каждого этапа g – тупиковый поток в $AN(f)$.*

Доказательство. Переход на новый этап происходит, когда в стр. 10 МАХП проверяется, что $cnc(s) = 0$ или $cnc(t) = 0$. В первом случае $g(e) = ac'(e)$ для всякой дуги e, выходящей из s, во втором - то же имеет место для всякой дуги e, входящей в t. В обоих случаях это означает, что в $AN(f)$ нет прямого увеличивающего пути.

Лемма 4. *Расстояние от s до t в $AN(f + g)$ на произвольном этапе строго больше, чем расстояние от s до t в $AN(f)$ на предыдущем этапе.*

Доказательство. Вспомогательная сеть $AN(f + g)$ совпадает со вспомогательной сетью сети $AN(f)$ относительно потока g (Доказать!). По лемме 3 поток g – тупиковый, т.е. в $AN(f)$ нет прямых увеличивающих путей относительно g. Поэтому все имеющиеся увеличивающие пути "непрямые", т.е. их длины больше, чем расстояние от s до t в $AN(f)$, и лемма следует из теоремы 4.б.

Теорема 5. *Алгоритм МАХП корректно решает задачу о максимальном потоке для сети $N = (V, E, s, t, c)$ за время $O(n^3)$ $(n = |V|)$.*

Доказательство. МАХП завершается (стр. 6), когда в сети $AN(f)$ s и t не связаны. Но тогда в этой сети максимальный поток нулевой и по теореме 4.в $val(f) = max f$, т.е. f – максимальный поток.

Для оценки времени отметим, что по лемме 4 число этапов не превосходит $|V|$. На каждом этапе стр. 4 и 5 требуют $O(|V|^2)$ шагов для инициализации $AN(f)$. После выполнения процедур ПРОТОЛКНУТЬ и ПРОТЯНУТЬ для вершины v ее потенциал $cnc(v)$ становится нулевым *(почему ?)* и она удаляется из сети (стр. 11). Поэтому число вызовов каждой из них $< |V|$. В процедуре ПРОТОЛКНУТЬ каждая вершина может попасть в очередь Q только 1 раз, поэтому сложность этой процедуры – $O(|V|)$. Таким образом, каждый этап можно выполнить за время $O(|V|^2)$, а общее время работы алгоритма МАХП – $O(|V|^3)$.

ЗАДАЧА. Написать алгоритм для процедуры ПРОТЯНУТЬ.

1.1 СЕТИ С ЕДИНИЧНЫМИ ПРОПУСКНЫМИ СПОСОБНОСТЯМИ

Рассмотрим работу алгоритма МАХП на сетях, в которых пропускные способности всех дуг равны 1.

Лемма 5. *Каждый этап алгоритма МАХП, примененного к сети* $N = (V, E, s, t, c)$, *такой что* $c(e) = 1$ *для любой дуги e из E, можно выполнить за* $O(|E|)$ *шагов.*

Доказательство. В этом случае и во вспомогательной сети $AN(f)$ на каждом этапе пропускные способности дуг будут равны 1. Но тогда кажда дуга может обрабатываться в процедурах ПРОТОЛКНУТЬ и ПРОТЯНУТЬ не более 1 раза, после чего она "насыщается"и удаляется из сети.

Лемма 6. *В сети* N *из леммы 5 расстояние* l *между* s *и* t *не может быть больше, чем* $2 * |V|/\sqrt{val(fm)}$ *, где* fm *– максимальный поток.*

Доказательство. Пусть $V_i = \{v \mid$ расстояние от s до v равно $i\}$. Множество вершин $VV_i = V_0 \cup V_1 \cup ... \cup V_i$ задает разрез в N при $i < l$. Тогда $f(VV_i, V \setminus VV_i) = |\{e \mid e$ – дуга из V_i в $V_{i+1}\}| \geq val(fm)$. Но число дуг из V_i в V_{i+1} не больше, чем $|V_i| * |V_{i+1}|$. Поэтому либо $|V_i| \geq \sqrt{val(fm)}$, либо $|V_{i+1}| \geq \sqrt{val(fm)}$. Отсюда следует, что по крайней мере для каждого второго уровн V_i имеет место $|V_i| \geq \sqrt{val(fm)}$ и $l * \sqrt{val(fm)} \leq 2 * |V|$.

Теорема 6. *Для сетей с единичными пропускными способностями дуг время работы алгоритма МАХП не превосходит* $O(|E| * |V|^{2/3})$.

Доказательство. Покажем, что число этапов s не превосходит $|V|^{2/3}$.

а) Если $val(fm) \leq |V|^{2/3}$, то этапов "мало", так как на каждом этапе поток увеличивается по крайней мере на 1.

б) Если $val(fm) > |V|^{2/3}$, то пусть i – такой этап, после которого впервые поток превосходит $val(fm) - |V|^{2/3}$. Пусть g – это поток в начале этапа i. Так как величина дополнительного потока $\leq |V|^{2/3}$, то $s - i \leq |V|^{2/3}$. С другой стороны, $i <$ (расстояние от s до t в сети $AN(g)$) $\leq 2 * |V|/\sqrt{val(fm) - val(g)}$ (по лемме 6 и так как величина максимального потока в $AN(g)$ равна $val(fm) - val(g)$ по теореме 4.в). Но по выбору g имеем перед i-ым этапом $val(fm) - val(g) \geq |V|^{2/3}$. Отсюда $i \leq 2 * |V|^{2/3}$ и $s \leq 3 * |V|^{2/3}$. Теперь теорема следует из леммы 5 и этой оценки числа этапов.

Определение 8. *Сеть, у которой пропускные способности всех дуг равны 1, назовем* простой, *если для любой вершины сети степень захода или степень исхода равны 1 или 0.*

Лемма 7. *В простой сети* N *расстояние* l *между* s *и* t *не может быть больше, чем* $|V|/val(fm)$, *где* fm *– максимальный поток.*

Доказательство. Определим V_i как в лемме 6. Так как через каждую вершину может проходить не более одной единицы максимального потока, а весь он проходит через каждое V_i, то $val(fm) \leq |V_i|$. Отсюда $l * val(fm) \leq |V|$.

Теорема 7. *Для простых сетей время работы алгоритма МАХП не превосходит* $O(|E|\sqrt{|V|})$.

Доказательство. Покажем, что число этапов s не превосходит $\sqrt{|V|}$.

а) Если $val(fm) \leq \sqrt{|V|}$, то этапов "мало", так как на каждом этапе поток увеличивается по крайней мере на 1.

б) Если $val(fm) > \sqrt{|V|}$, то пусть i – такой этап, после которого впервые поток превосходит $val(fm) - \sqrt{|V|}$. Пусть g – поток в начале этапа i. Величина максимального потока в $AN(g)$ $\geq \sqrt{|V|}$. Тогда по лемме 7 $l \leq \sqrt{|V|}$. (Докажите, что $AN(f)$ – простая сеть и лемма применима). Поэтому $i \leq \sqrt{|V|}$ и $s - i \leq \sqrt{|V|}$, т.е. $s = O(\sqrt{|V|})$.

1.2 ПАРОСОЧЕТАНИЯ В ДВУДОЛЬНЫХ ГРАФАХ

Определение 9. Паросочетание *в неориентированном графе* $G = (V, E)$ *– это произвольное подмножество ребер* M *такое, что никакие две дуги из него не имеют общей вершины. Паросочетание наибольшей мощности называется максимальным.*

Определение 10. *Граф* $G = (V, E)$ *называется* двудольным, *если множество его вершин* V *можно разбить на два непересекающихся подмножества* X *и* Y *так, что каждое ребро* $e \in E$ *имеет вид* $e = (x, y)$, *где* $x \in X$, *а* $y \in Y$.

Пусть дан двудольный граф $B = (X, Y, E)$. Определим сеть $N(B) = (W, A, s, t, c)$:

1) вершины – $W = s, t \cup X \cup Y$;

2) дуги – $A = \{(s, x) \mid x \in X\} \cup \{(y, t) \mid y \in Y\} \cup \{(x, y) \mid (x, y) \in E\}$;

3) пропускные спсобности $c(e) = 1$ для всех $e \in A$.

Замечание. Построенная сеть $N(B)$ - простая.

Лемма 8. *Мощность максимального паросочетания в графе* B *равна величине максимального потока в сети* $N(B)$.

Доказательство. Пусть M – паросочетание в B. Определим поток f в $N(B)$ следующим образом. Для каждой дуги $(u, v) \in M$ положим $f(s, u) = 1$, $f(u, v) = 1$, $f(v, t) = 1$. Тогда f - допустимый поток и $val(f) = |M|$.

Пусть теперь f – максимальный поток в $N(B)$, построенный алгоритмом МАХП. Тогда он имеет целочисленные значения (0 или 1). Пусть M состоит из таких дуг (u, v), для которых $f(u, v) = 1$. Нетрудно проверить, что M – паросочетание и что $|M| = val(f)$.

Теорема 8. *Задачу о максимальном паросочетании для двудольных графов можно решить за время* $O(|E| * \sqrt{|V|})$.

Доказательство. Алгоритм поиска максимального паросочетания работает следующим образом.

1) По графу B строим сеть $N(B)$.

2) С помощью алгоритма МАХП находим максимальный поток в $N(B)$.

3) По этому потоку строим максимальное паросочетание в B (по лемме 8).

Так как $N(B)$ – простая сеть, то время 2-го этапа - $O(|E| * \sqrt{|V|})$. Время 1-го этапа – $O(|E|)$, а время 3-го – $O(|V|)$ (п о ч е м у ?). Отсюда следует оценка теоремы.