Оглавление

Глава 1

Графы

1.1 Продолжение про деревья

Алгоритм (Прима).

1. Выбираем $i \in M$
   $\text{num}(i) := 0$
2. Рассмотрим рёбра (u, v) : помечена ровно одна из вершин
   $\overline{N}$ – кайма
3. $l(\overline{e}) = \min\limits_{l \in \overline{N}} l(e)$
4. Добавляем $\overline{e}$ к решению
   $N' = N' \cup \{\overline{e}\}$

Теорема 1. Алгоритм Прима строит оптимальное остовное дерево

Теорема 2. Следующие определения дерева эквивалентны:

1. Связный граф без циклов
2. Максимальный граф без циклов[a]
3. Любые две вершины соединены единственной цепью
4. Минимальный связный граф[b]
5. Связный граф: $|N| = |M| - 1$
6. Граф без циклов: $|N| = |M| - 1$

[a] Если добавить ребро между любыми двумя вершинами, то появится цикл
[b] Если убрать любое ребро, то связность пропадёт

Доказательство.

- $1 \implies 2$
  Если в связный граф добавить ребро, то он замкнётся
- $2 \implies 3$
  Граф **максимальный** без циклов, значит, любые две вершины соединены (если добавить ребро, они будут соединены двумя цепями)
  Отсюда же следует единственность
- $3 \implies 4$
  Любые две вершины соединены цепью $\implies$ граф связный
  Минимальность следует из единственности
- $4 \implies 5$

- $|N| > |M| - 1 \implies$ есть цикл
- $|N| < |M| - 1 \implies$ граф не связный

- $5 \implies 6$
- $6 \implies 1$

□

1.2 Реализация алгоритмов Прима и *Крастера*

Алгоритм (*Крастера*). Будем хранить промежуточный результат в виде компонент связности

- Сортировка: $l(e_1) \leq l(e_2) \leq ... \leq l(e_n)$
- ```
  for e = (u, v) \in E do
      if FindSet(u) != FindSet(v)
          then {A := A \cup {e}, Union(u, v)}
  ```

- Процедура Union:

```
p(x) := x       # родитель вершины x
   define Link(x, y) {
       if rank(x) > rank(y)
           then p(y) := x
           else p(x) := y
       if rank(x) = rank(y)
           then rank(y) := rank(x) + 1
   define Union(u, v) = Link(FindSet(u), FindSet(u))
```
```