

Минимальные остовные деревья. Алгоритмы Крускала и Прима.

Работу выполнил
Студент группы
И-2-10
Бекиров Алим

Минимальные остовные деревья

Задача: в процессе разработке электронных схем необходимо получить такую компоновку из $n-1$ проводов, которая использует минимальное количество провода.

Дан некий связный неориентированный граф $G=(V,E)$, где V - множество контактов, E - множество возможных соединений между парами контактов, и для каждого ребра $(u,v) \in E$ задан вес $w(u,v)$, определяющий стоимость (количество необходимого провода) соединения u и v .

Минимальные остовные деревья

Необходимо найти ациклическое подмножество T , которое соединяет все вершины и чей общий вес минимален.

$$w(T) = \sum_{(u,v) \in T} w(u, v)$$

Остовным деревом (spanning tree)- называется множество T , которое ациклично и связывает все вершины, при этом данное дерево имеет минимальный вес.

Задачей поиска минимального остовного дерева (minimum spanning tree problem) называют задачей поиска дерева T .

Минимальные остовные деревья

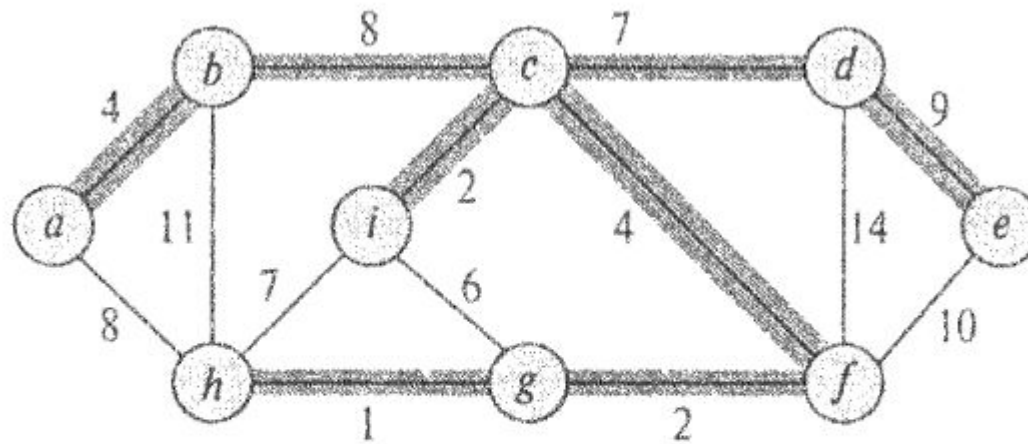


Рис. Минимальное остовное дерево
связного графа

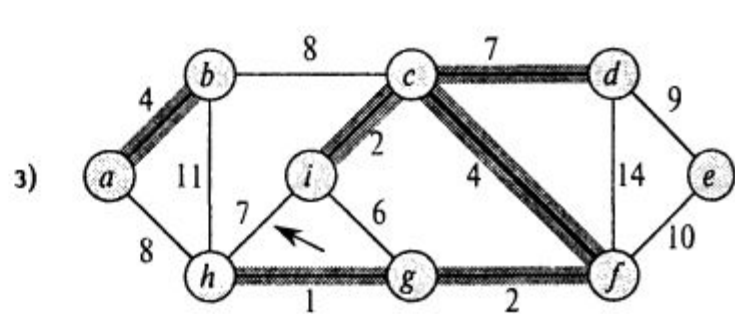
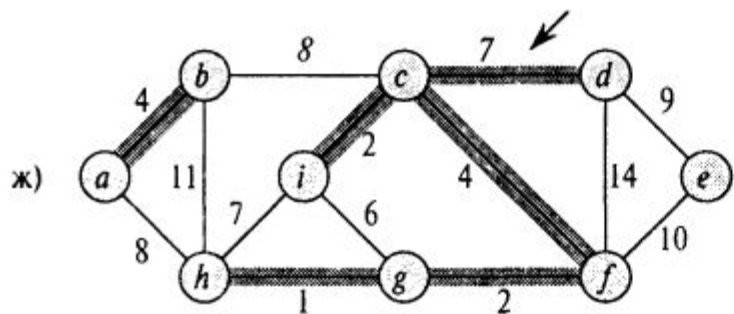
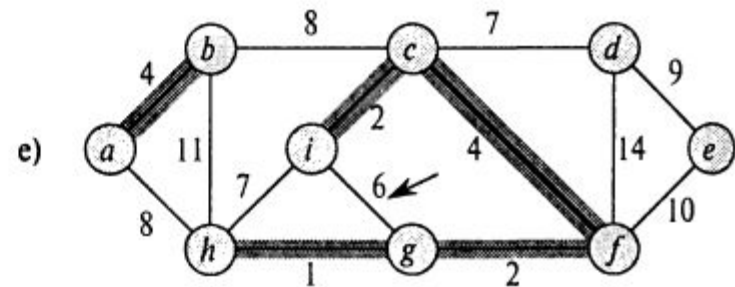
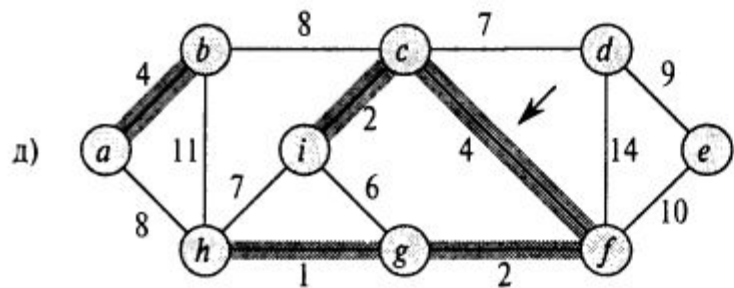
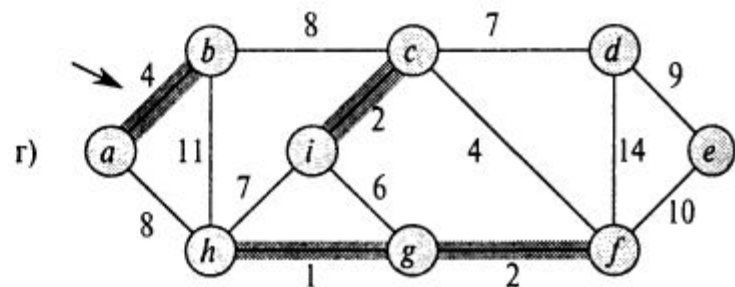
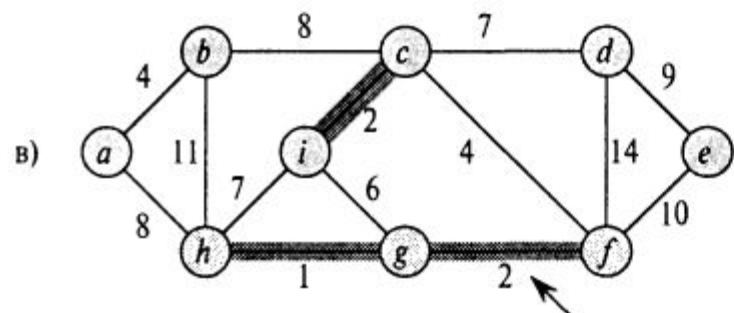
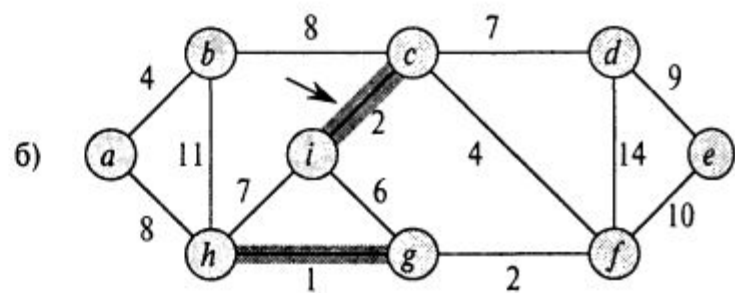
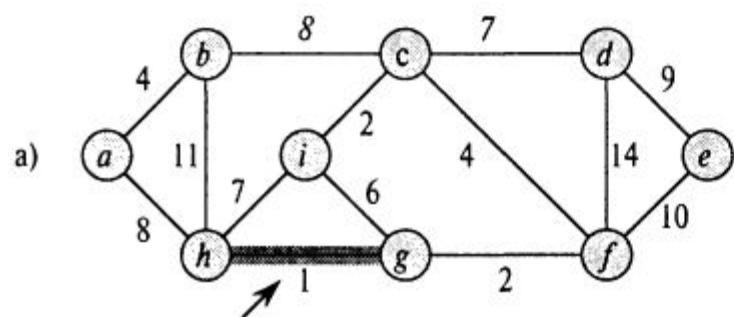
Алгоритм Крускала

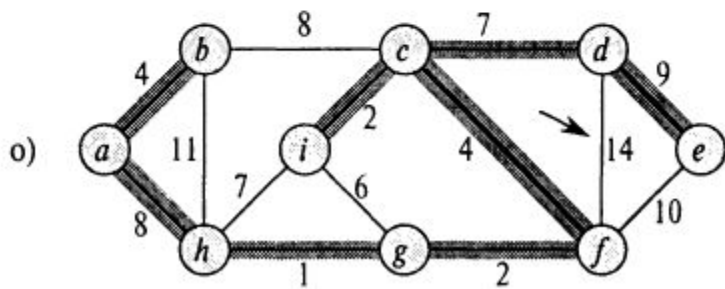
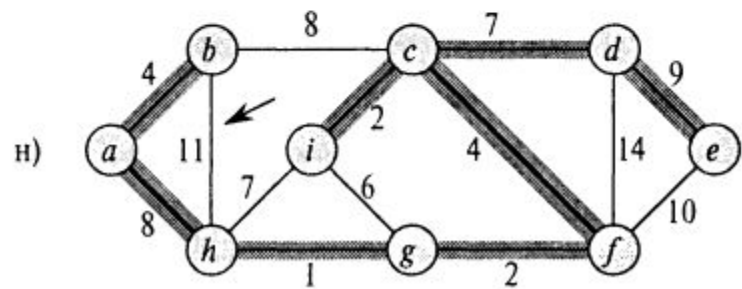
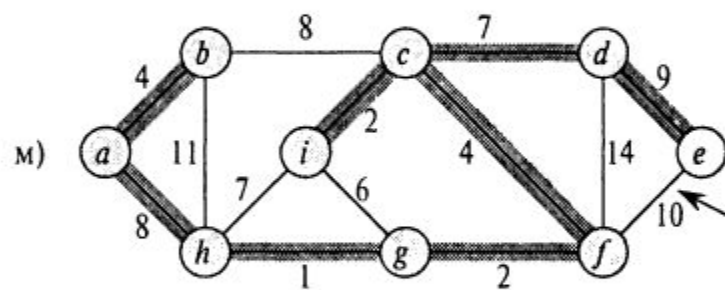
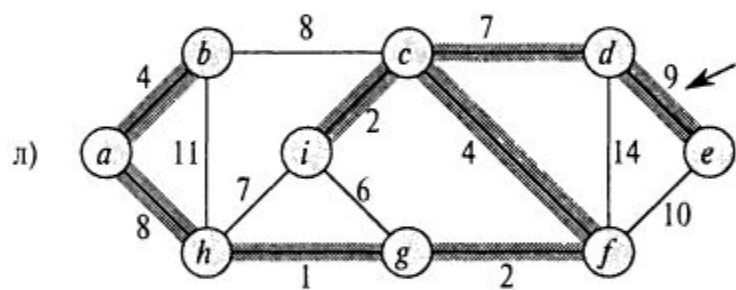
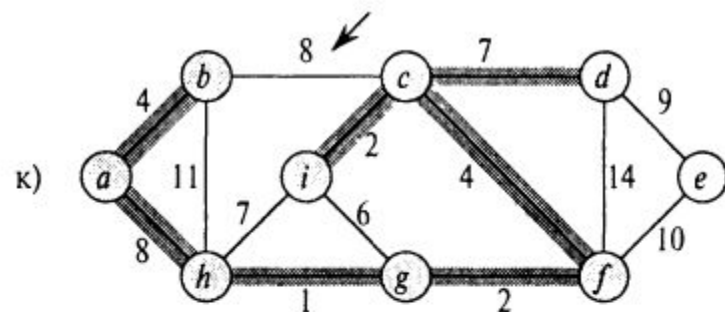
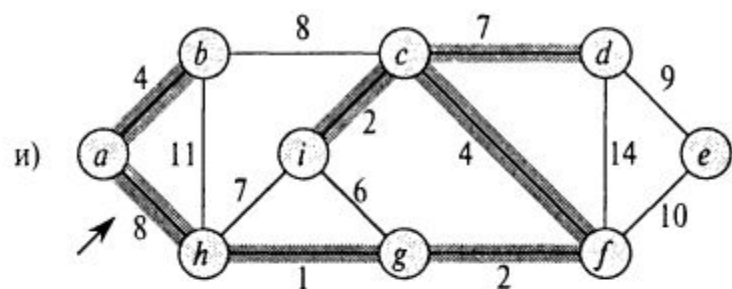
- Алгоритм Крускала предназначен для нахождения безопасного ребра для добавления в растущий лес путем поиска ребра (u, v) с минимальным весом среди всех ребер, соединяющий два дерева в лесу.
- Данный алгоритм является жадным, поскольку на каждом шаге он добавляет к лесу ребро с минимально возможным весом.
- Время работы алгоритма $O(E \lg V)$.

Алгоритм Крускала

MST_KRUSKAL(G, w)

```
1   $A \leftarrow \emptyset$ 
2  for (Для) каждой вершины  $v \in V[G]$ 
3      do MAKE_SET( $v$ )
4  Сортируем ребра из  $E$  в неубывающем порядке их весов  $w$ 
5  for (Для) каждого  $(u, v) \in E$  (в порядке возрастания веса)
6      do if FIND_SET( $u$ )  $\neq$  FIND_SET( $v$ )
7          then  $A \leftarrow A \cup \{(u, v)\}$ 
8              UNION( $u, v$ )
9  return  $A$ 
```





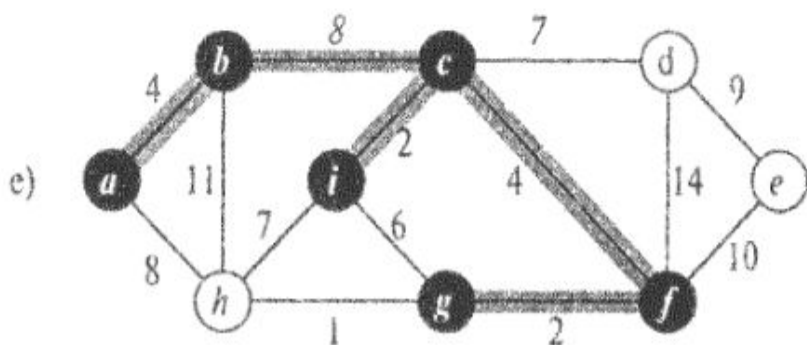
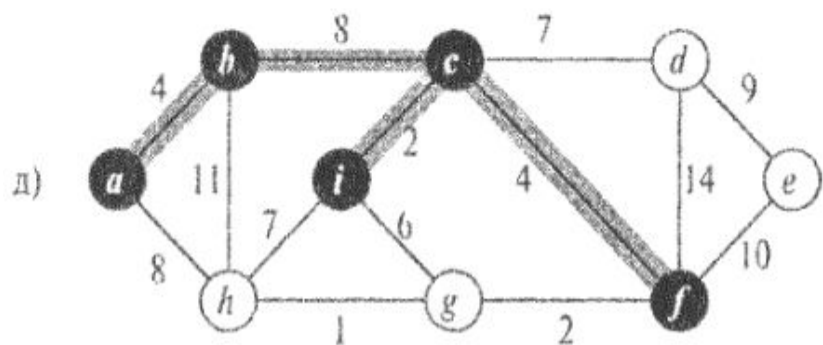
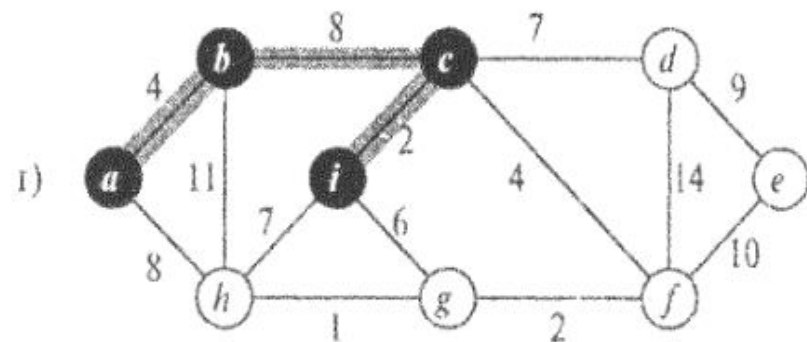
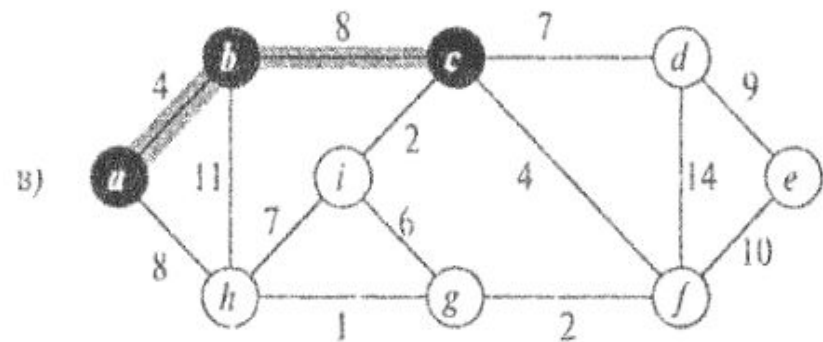
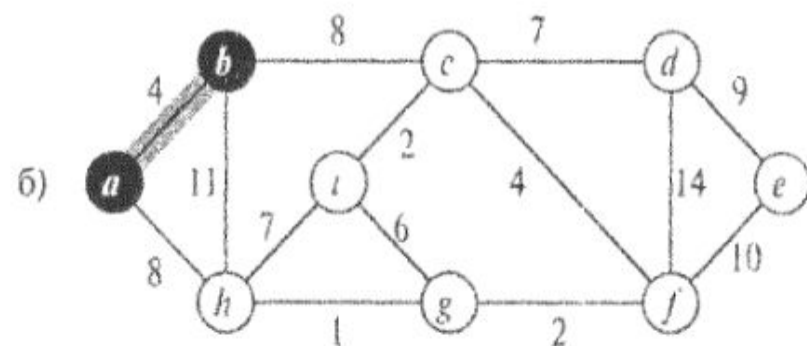
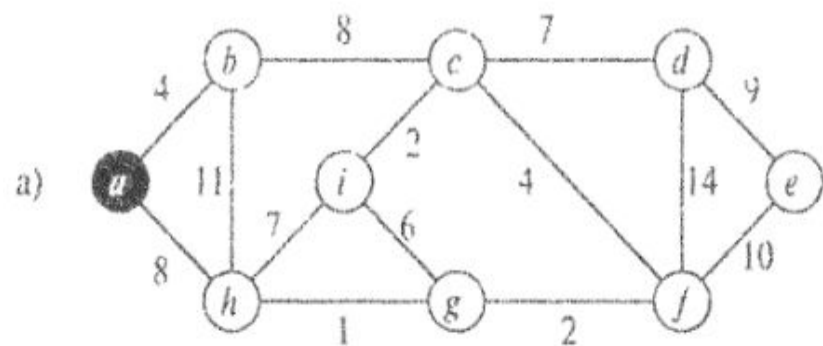
Алгоритм Прима

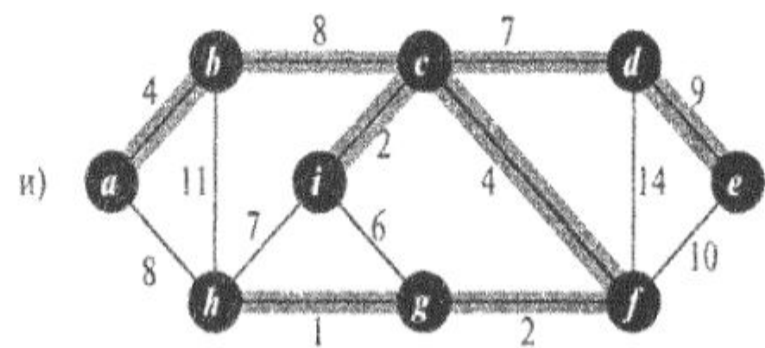
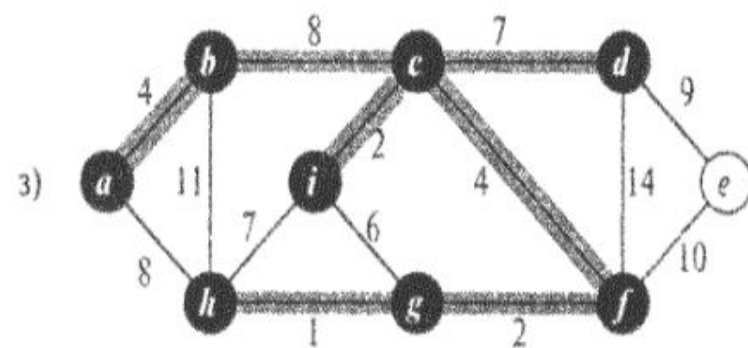
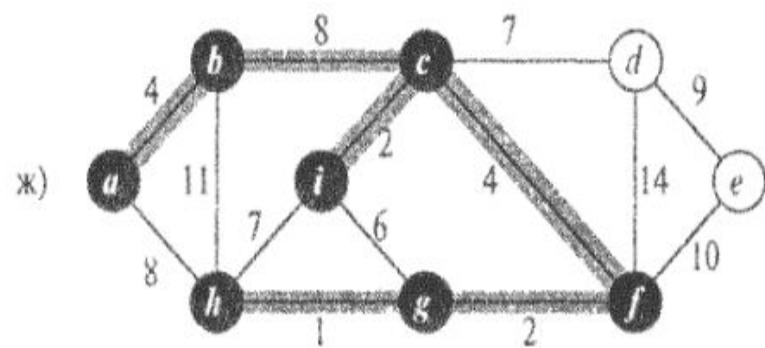
- Аналогично алгоритму Крускала. Построение дерева начинается с произвольной корневой вершиной r и растет до тех пор пока не охватит все вершины в V . На каждом шаге к дереву A добавляется легкое ребро, соединяющее дерево и отдельную вершину из оставшейся части графа.
- Время работы алгоритма $O(E + V \lg V)$.

Алгоритм Прима

MST_PRIM(G, w, r)

```
1  for (Для) каждой вершины  $u \in V[G]$ 
2      do  $key[u] \leftarrow \infty$ 
3       $\pi[u] \leftarrow \text{NIL}$ 
4   $key[r] \leftarrow 0$ 
5   $Q \leftarrow V[G]$ 
6  while  $Q \neq \emptyset$ 
7      do  $u \leftarrow \text{EXTRACT\_MIN}(Q)$ 
8      for (Для) каждой вершины  $v \in Adj[u]$ 
9          do if  $v \in Q$  и  $w(u, v) < key[v]$ 
10             then  $\pi[v] \leftarrow u$ 
11              $key[v] \leftarrow w(u, v)$ 
```





Спасибо за внимание!