

Кратчайшие пути в графах

Лекция 10

Обход в ширину (BFS)

- $O(|E|)$
- Невзвешенные ребра
- Одна компонента связности

```
bfs(V, E, s)
    used[s] = true
    queue.push(s)
    while not queue.isEmpty()
        v = queue.pop()
        for (v, u) in E
            if not used[u]
                used[u] = true
                queue.push(u)
```

Обход в ширину

- Если хотим считать расстояние

```
dfs(V, E, s)
    d[s] = 0
    used[s] = true
    queue.push(s)
    while not queue.isEmpty()
        v = queue.pop()
        for (v, u) in E
            if not used[u]
                used[u] = true
                queue.push(u)
                d[u] = d[v] + 1
```

Алгоритм Дейкстры

- $O(|E|^2 + |V|)$
- Не работает с отрицательными весами
- Можно ускорить, если использовать приоритетную очередь

```
for v in V
    d[v] = INF
d[s] = 0
for i = 0 to |V| - 1
    next = -1
    for v in V
        if not used[v] and
            (next == -1 or d[v] < d[next])
            next = v
    used[next] = true
    for (next, u) in E
        d[u] = min(d[u],
            d[next] + W(next, u))
```

Алгоритм Форда-Беллмана

- $O(|E||V|)$
- Работает с отрицательными весами

```
for v in V
    d[v][0] = INF
d[s][0] = 0
for k = 0 to |V| - 1
    for (u, v) in E
        d[v][k] = min(d[v][k],
                       d[u][k - 1] + W(u, v))
```

Алгоритм Форда-Беллмана

```
for v in V
    d[v] = INF
d[s] = 0
for k in range(0, |V|)
    for (u, v) in E
        d[v] = min(d[v],
                    d[u] + W(u, v))
```

- Теперь память $O(|V|)$
- Поддерживаем минимум в одномерном массиве

Цикл отрицательного веса

- Находим вершину, достижимую из цикла
- Если получилось обновить веса, то есть цикл отрицательного веса

```
for (u, v) in E
    if d[v] > d[u] + W(u, v)
        # есть цикл
```

Цикл отрицательного веса

- Создаем массив p , в котором храним путь
- Для вершин в цикле верно, что $p[v_{\{i\}}] = v_{\{i-1\}}$
- Если сделаем v шагов назад, то попадем в цикл

```
for (u, v) in E
    if d[v] > d[u] + W(u, v)
        for i = 0 to |V| - 1
            v = p[v]
        cur = v
        while p[cur] != v
            ans.add(cur)
            cur = p[cur]
        ans.add(cur)
        ans = ans[::-1]
        break
```