2019/6/24 bfs01.hpp

```
#include "../../bits/stdc++.h"
#include "../graph.hpp"
 // verified: https://atcoder.jp/contests/abc126/submissions/5485170 std::vector<int> bfs01(const Graph &g, int s)
           int n = g.size();
std::vector<int> d(n, INF);
std::deque<int> que;
que.push_back(s);
d[s] = 0;
while (!que.empty())
{
 8
 9
10
11
12
13
                  int cur = que.front();
que.pop_front();
14
15
16
                  for (auto p : g[cur])
17
                         int dst = p.to, w = p.cost;
assert(w == 0 || w == 1);
if (d[dst] != INF)
18
19
20
21
                               continue;
                         if (w == 0)
22
23
                         {
                               d[dst] = d[cur];
que.push_front(dst);
24
25
26
                         else
27
28
                         {
                                d[dst] = d[cur] + 1;
29
                                que.push_back(dst);
30
31
32
                  }
33
34
            return d;
35 }
36
```

localhost:4649/?mode=clike 1/1