1、

① using aggregate analysis:

$$\sum_{i=1}^{n} c_i = \sum_{i=0}^{\lfloor \lg n \rfloor} 2^i + 3(n - \lfloor \lg n \rfloor - 1)$$
$$< 2^{1+\lg n} - 1 + 3n$$
$$= 5n - 1$$
$$= O(5n)$$

So the amortized cost per operation is $O(5n)/n = O(1)$

② using Accounting method:

Charge 5 for each operation ($\hat{c}_i = 5$). If i is not a power of 2, pay 1 from credit.

If i is a power of 2, pay i from credit

We have proved that $\sum_{i=1}^{n} c_i < 5n-1 < 5n = \sum_{i=1}^{n} \hat{c}_i$

such a amortized cost is valid, For example

each operation costs $5 = O(1)$ on average.

③ using potential method:

定义一个势能函数.

$$\Phi(D_i) = \begin{cases} 0, & i = 0 \\ 2i - (2^{1+\lfloor \lg i \rfloor} - 1), & i \geq 1 \end{cases}$$

如果i不是2的幂,那么 $\hat{c}_i = c_i + \Phi_i(D_i) - \Phi(D_{i-1})$

$$= 3 + 2 - 2^{1+\lfloor \lg i \rfloor} + 2^{1+\lfloor \lg (i-1) \rfloor}$$
$$= 5$$

如果i是2的幂,例如对于 $i = 2^j$ ($j \in \{0,1,2 \cdots\}$),有:

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

华东师范大学
EAST CHINA NORMAL UNIVERSITY

$= i+2 - 2^{i+j} + 2^j = 2$

因为 $\hat{c}_i \le 5$，每个操作的平均代价为 $O(1)$

22.1-6

Assume adjacency matrix is M,

~~依~~ Vertex k is a universal sink iff the k-th row of M is 0 and the k-th column of M is 1.

Scan M from M[1,1]

① M[i,j]=1, k>i, increase i;

② M[i,j]=0, i≠j, k>j, increase j

③ M[i,j]=0, i>j, i is the potienal target row, increase j to check whether the row is 0.

Scan:

```
i=j=1
while i<|V| and j<|V|
    if M[i,j]=1
        i=i+1
    else if M[i,j]=0
        j=j+1
```

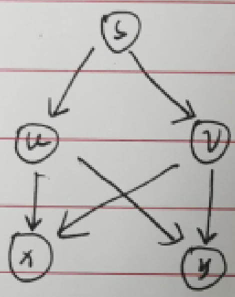
If there exists a universal sink, it must be vertex i, add a check in the end

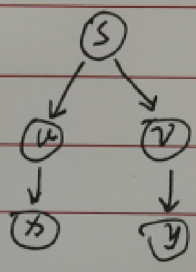
```
CONTAIN-UNIVERSAL-SINK(M)
  i=j=1
  while i<|V| and j<|V|
    if M[i,j]=1
      i=i+1
    else if M[i,j]=0
      j=j+1
  for j=1 to |V|
    if j≠i and M[i,j]=0
                                      return false
                                    return true
```

闵行校区：中国·上海市东川路500号 邮编：200241 500 DongChuan Rd, Shanghai 200241, P.R.China
普陀校区：中国·上海市中山北路3663号 邮编：200062 3663 Zhongshan Rd(N), Shanghai 200062, P.R.China
http://www.ecnu.edu.cn Tel:(86-21)62233333

Scan costs $O(V)$, check costs $O(V)$. The algorithm costs $O(V)$

22.2-6



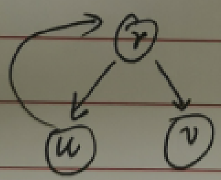


The tree

The G

The shortest path from s to v is same, but BFS never generate such a tree

22.3-9



$\Rightarrow$ contains a loop back

Start DFS from r and visit u before v. $v.d = 4$, $u.f = 3$. $v.d \nleq u.f$, but there exits a path $u \rightarrow r \rightarrow v$