Alg HW7

1. (1) $c_i=\begin{cases} i, & i\text{为2的幂} \\ 1, & \text{else} \end{cases}$, $\sum_{i=1}^{n} c_i = \sum_{i=1}^{\lceil \lg n \rceil} 2^i + \text{else}$

$$\le \sum_{i=1}^{\lceil \lg n \rceil} 2^i + n \le 2\cdot 2^{\lceil \lg n \rceil} - 1 + n \in O(n)$$

$O(n)/n = O(1)$.

(2) 为每一次操作赋予摊还代价3.

$i=1$, 代价为1, 获得2点信用;

假定$i=2^k$时仍有信用. 从$i=2^k$到$i=2^{k+1}-1$, 我们至少获取$(3-1)(2^k-1)$

$=2^{k+1}-2$的信用, 到了$i=2^{k+1}$, 我们仍有$2^{k+1}-2+3-2^{k+1}=1$的信用, 即信用

可以始终非负. 故摊还代价为$O(1)$.

(3) $\Phi(D_i)=2i-2^{\lfloor \lg i \rfloor+1}$, $i>0$, $\Phi(D_0)=0$

此时, $\Phi(D_i)-\Phi(D_{i-1})=\begin{cases} 2-2^k, i=2^k, & \hat{c}_i=c_i+2-2^k=2 \\ 2, \text{else}, & \hat{c}_i=c_i+2=3 \end{cases} \Rightarrow O(1)$.

即摊还代价$O(1)$.

2. 使用2个栈, A.B. A为数据栈, B为MIN栈.

1° ENQUEUE: A入栈, 同时B处作一额外操作, 与栈顶元素比较, 若更小则入B栈, 否则B栈顶元素重复入栈;

2° DEQUEUE: A、B同时出栈.

3° FIND-MIN: 返回B栈顶元素.

可以看出, 三个操作复杂度均为$O(1)$, 显然其摊还代价亦为$O(1)$.

3. 贪心算法：每次选择能使当前选择覆盖程度最大的集合加入直到选择了k个.

证明：记 $f(N)=|\bigcup_{i\in N}S_i|$，记 O 为最优解集. 令 S_i 为下一时刻选择的集合.

有 $f(N\cup\{S_i\})-f(N)\geq\frac{1}{k}(f(O)-f(N))$，$N$ 为当前时刻解集

首先 N 为空集，那么必有 $f(S_i)\geq\frac{1}{k}f(O)$. 即 O 中必有集合收益不小于 $\frac{1}{k}f(O)$. 而这个集合恰好被贪心算法选择. 而在此之后，又必有某集合收益不小于 $\frac{1}{k}(f(O)-f(S_i))$，由此往复即有此式.

记 N_t 为 t 时刻解集，N_k 为最终贪心解. 有

$$f(N_k)-f(N_{k-1})\geq\frac{1}{k}(f(O)-f(N_{k-1}))$$

$$\begin{aligned}
\Rightarrow\quad f(N_k)&\geq\frac{1}{k}f(O)+\left(1-\frac{1}{k}\right)f(N_{k-1})\\
&\geq\frac{1}{k}f(O)+\left(1-\frac{1}{k}\right)\left(\frac{1}{k}f(O)+\left(1-\frac{1}{k}\right)f(N_{k-2})\right)\\
&\geq\frac{1}{k}f(O)\left(1+\left(1-\frac{1}{k}\right)+\left(1-\frac{1}{k}\right)^2+\cdots+\left(1-\frac{1}{k}\right)^{k-1}\right)\\
&=\left(1-\left(1-\frac{1}{k}\right)^k\right)f(O)\geq\left(1-\frac{1}{e}\right)f(O).
\end{aligned}$$

即近似比为 $1-\left(1-\frac{1}{k}\right)^k\geq1-\frac{1}{e}$.