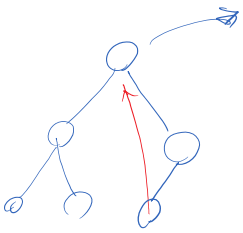
Midterm review

Tuesday, March 16, 2021 10:08 AM



$$\phi(D_i) = \sum_{j=1}^{s} \lg j$$

op	actual cost	$\Delta\phi$	Amortized cost
insert	$\lg s$	$\sum_{j=1}^{s} \lg j - \sum_{j=1}^{s-1} \lg i = \lg s$	$\lg s + \lg s = 2\lg s$
extract-min	$\lg s$	$\sum_{j=1}^{s} \lg j - \sum_{j=1}^{s-1} \lg j = -\lg s$	$\lg s - \lg s = 0$

```
1: function RAND-ALG(A, n)
2:     s = A[1]
3:     k = RANDOM(n)
4:     if k < log2 n then
5:         for i = 1 to n do
6:             j = 1
7:             while j < n do
8:                 s = s + A[i] * A[j]
9:                 j = j * 2
```

worst-case: $\Theta(n \cdot \lg n)$

Average case:

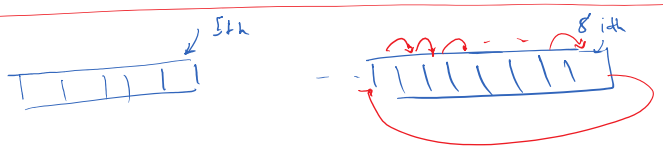
$$Pr\{k < \lg n\} = \frac{\lg n}{n}$$

$$E[T(n)] = Pr\{k < \lg n\} \cdot Time\{k < \lg n\} + Pr\{k \geq \lg n\} \cdot Time\{k \geq \lg n\}$$

$$= \frac{\lg n}{n}(c n \lg n + d) + \left(1 - \frac{\lg n}{n}\right) d$$

$$= c \lg^2 n + d$$

$$= \Theta(\lg^2 n)$$



$$c_i = \begin{cases} i+1 & \text{when } i \text{ is exact power of 2} \\ 1 & \text{otherwise} \end{cases}$$

n $\lg n$

1 otherwise

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

$$1 + 2 + 1 + 4 + 1 + 1 + 1 + 8$$

$\Downarrow$

avg cost of each op $< 3n/n = 3$

amortized cost $= O(1)$

accounting method: \$3 Cost for insert(i) → \$1 insertion, \$2 credit (when not power of two)

when i is exact power of 2, pay \$i using the stored credit

op	Cost	actual cost	credit remaining
1	3	1	2
2	3	2	3
3	3	1	5
4	3	4	4
.	.	.	.
.	.	.	.

Course selection DP

choice: Consider choosing or not choosing the last course

$t[i, m]$ = max value of credits taken in first i courses while total credits $\leq m$

recursive formula:

$$t[i,m] = \begin{cases} 0 & \text{if } i = 0 \text{ or } m = 0 \quad \text{(no more courses left)} \\ t[i-1, m] & \text{if } c[i] > m \quad \text{(course doesn't fit in schedule)} \\ \max\left(t[i-1, m],\ v[i] + t[i-1, m - c[i]] \right) & \text{if } c[i] \leq m \ (i \neq 0 \ \& \ m \neq 0 \ \&) \end{cases}$$

$t[i-1, m]$: (choice: don't take course i)

$v[i] + t[i-1, m-c[i]]$: (choice: take course i)

Recursive Alg:

Best Value Plan (c, v, n, M) {

```
    for i=1 to n
        for m=1 to M
            t[i,m] = -∞

    return BVP_Aux(c, v, n, M, t)
}

BVP_Aux(c, v, n, M, t){
    if n==0 or M==0
        return 0
    if t[n,M] != -∞
        return t[n,M]
    if c[n] > M
        t[n,M] = BVP_Aux(c, v, n-1, M, t)
    else
        t[n,M] = max(BVP_Aux(c,v, n-1, M, t), v[n]+BVP_Aux(c, v, n-1, M-c[n], t))

    return t[n,M]
}
```