CSE 102 10-28-21

- supplemental lecture tomorrow
- hw 4 Posted
- Programming Project

Average case runtime of Quicksort:

- our assumption that all $n!$ permutations of $A[1 \cdots n]$ are eq. likely implies that pivot $A[n]$ is equally likely to be placed in any one of n slots

$$A[1 \cdots\cdots\cdots n]$$
$$\uparrow$$
$$A[q]$$

i.e. i.e. $P(q=i) = \frac{1}{n}$ (for $i=1,\dots n$)

note

$$\text{length}\left(A[1 \cdots (q-1)]\right) = q-1$$

and

$$\text{length}\left(A[(q+1) \cdots n]\right) = n-q .$$

thus

$$t(n) = \frac{\sum_{q=1}^{n} \left((n-1) + t(q-1) + t(n-q) \right)}{n}$$

so

$$t(n) = (n-1) + \frac{1}{n} \sum_{q=1}^{n} \left(t(q-1) + t(n-q) \right)$$

observe: $\underbrace{t(0)=0}$, $t(1)=0$, $t(2)=1$.

$$t(n) = (n-1) + \frac{1}{n} \left(\sum_{q=1}^{n-1} t(q) + \sum_{q=1}^{n-1} t(n-q) \right)$$

$$\boxed{t(n) = (n-1) + \frac{2}{n} \cdot \sum_{q=1}^{n-1} t(q)} \qquad (*)$$

To solve this, set

$$x_n = \sum_{q=1}^{n-1} t(q) \qquad \left(\begin{array}{l} \text{note:} \\ x_1 = 0 \end{array} \right)$$

then

$$x_{n+1} - x_n = \sum_{q=1}^{n} t(q) - \sum_{q=1}^{n-1} t(q) = t(n)$$

our recurrence (*) becomes

$$\boxed{x_{n+1} - x_n = (n-1) + \frac{2}{n} \cdot x_n}$$

$$\therefore \quad x_{n+1} - \left(1 + \frac{2}{n}\right) x_n = n-1$$

$$x_{n+1} - \left(\frac{n+2}{n}\right) x_n = n-1$$

multiply by : $\dfrac{1}{(n+1)(n+2)}$

$$\therefore \quad \frac{x_{n+1}}{(n+1)(n+2)} - \frac{x_n}{n(n+1)} = \frac{n-1}{(n+1)(n+2)}$$

$$= \frac{3}{n+2} - \frac{2}{n+1}$$

change n to k:

$$\frac{x_{k+1}}{(k+1)(k+2)} - \frac{x_k}{k(k+1)} = \frac{3}{k+2} - \frac{2}{k+1}$$

sum for $k = 1$ to $(n-1)$:

$$\sum_{k=1}^{n-1} \left(\frac{x_{k+1}}{(k+1)(k+2)} - \frac{x_k}{k(k+1)} \right) = \sum_{k=1}^{n-1} \left(\frac{1}{k+2} + \frac{2}{k+2} - \frac{2}{k+1} \right)$$

$$= \sum_{k=3}^{n+1} \frac{1}{k} + \sum_{k=3}^{n+1} \frac{2}{k} - \sum_{k=2}^{n} \frac{2}{k}$$

so

$$\sum_{k=2}^{n} \frac{x_k}{k(k+1)} - \sum_{k=1}^{n-1} \frac{x_k}{k(k+1)}$$

$$= \sum_{k=3}^{n+1} \frac{1}{k} + \sum_{k=3}^{n+1} \frac{2}{k} - \sum_{k=2}^{n} \frac{2}{k}$$

$$\therefore \quad \frac{x_n}{n(n+1)} - \cancelto{0}{\frac{x_1}{2}} = \sum_{k=3}^{n+1} \frac{1}{k} + \frac{2}{n+1} - 1$$

$$= \sum_{k=1}^{n} \frac{1}{k} + \frac{1}{n+1} - 1 - \frac{1}{2} + \frac{2}{n+1} - 1$$

$$= \underbrace{\sum_{k=1}^{n} \frac{1}{k}}_{H_n \ (n^{th} \text{ harmonic } \#)} + \frac{3}{n+1} - \frac{5}{2}$$

$$\frac{x_n}{n(n+1)} = H_n + \frac{3}{n+1} - \frac{5}{2}$$

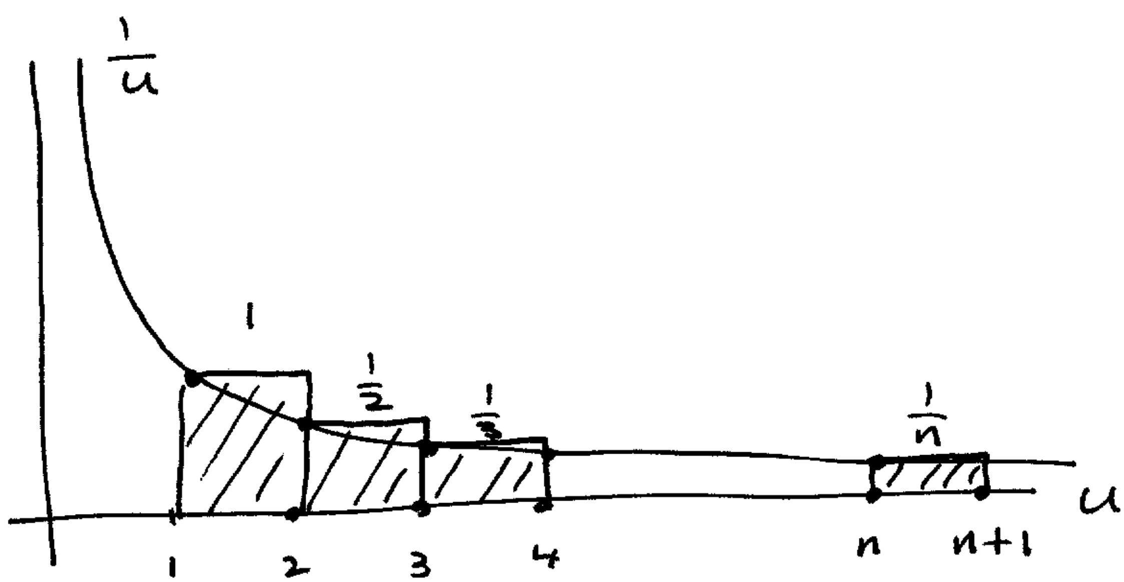
$\therefore \quad x_n = n(n+1)H_n + 3n + \frac{5}{2}n(n+1)$

$\therefore \quad t(n) = (n-1) + \frac{2}{n} x_n$

$$= (n-1) + 2(n+1)H_n + 6 - 5(n+1)$$

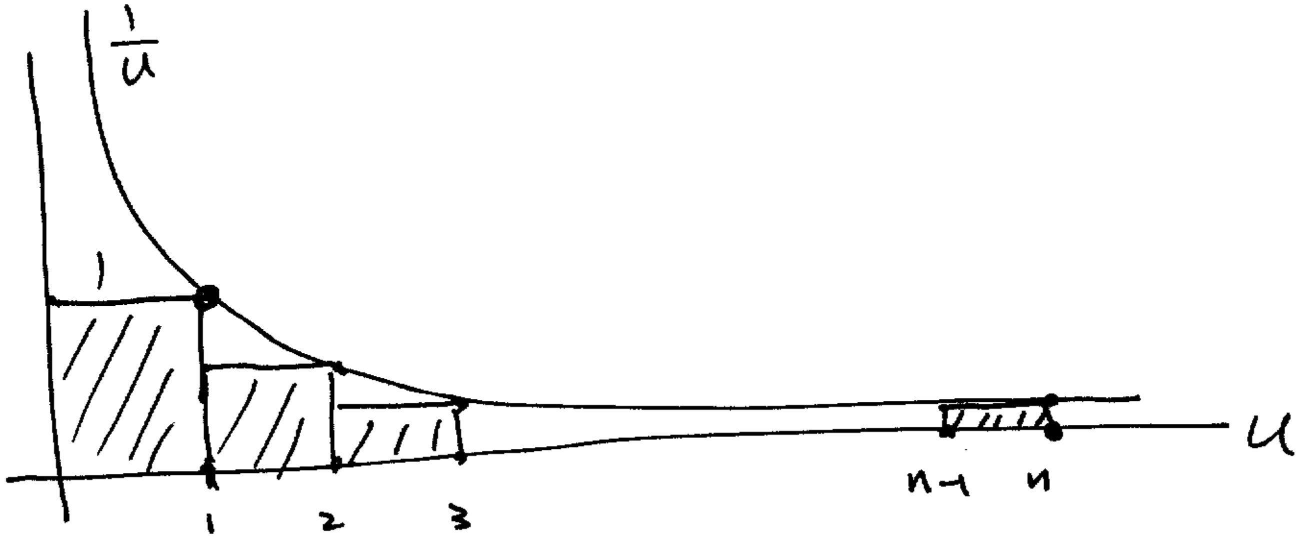
$\therefore \quad \boxed{t(n) = 2(n+1)H_n - 4n} \leftarrow$ exact soln.

estimate growth rate of H_n.



so

$$\int_1^{n+1} \frac{1}{u}\,du \le H_n \le 1 + \int_1^{n} \frac{1}{u}\,du$$



so

$$\underbrace{\ln(n+1)}_{\Omega(\ln(n))} \le H_n \le \underbrace{1 + \ln(n)}_{O(\ln(n))}$$

$\therefore \quad H_n = \Theta(\ln(n))$.

Actually $H_n \sim \ln(n)$

$$\therefore\ t(n) = 2(n+1)H_n - 4n$$

$$= 2n H_n + 2H_n - 4n$$

$$\sim 2n \ln n + 2 \ln n - 4n$$

$$\sim 2n \ln n$$

Hence

$$\boxed{t(n) = \Theta(n \log n)}$$ ← asympt. soln

Randomized Quicksort:

<u>RandPartition(A, p, r)</u>

1. i = Random(p, r)
2. A[i] ⟷ A[r] # swap
3. return Partition(A, p, r)

RandQuickSort(A, p, r)

1. if $p < r$
2. q = RandPartition(A, p, r)
3. RandQuicksort(A, p, q-1)
4. RandQuicksort(A, q+1, r)

Question:

Is it possible to do better than $\Theta(n \log n)$ (either worst or average case)?

Answer: not if basic op. is array comparisons.

(comparison based sorting.)

chap. 9 selection

Ex. minmax(A, p, r) finds **both**

min & max value in A[p...r]

use **min(m_1, m_2)** **max(M_1, M_2)**

1 comparison each.

```
if m1 < m2
    return m1
return m2
```

```
:
:
:
```

minmax(A, p, r) pre $p \le r$

```
if p = r
    return (A[p], A[p])
else
    q = ⌊(p+r)/2⌋
    (m1, M1) = minmax(A, q+1, r)
    (m2, M2 = minmax(A, p, q)
    return (min(m1, m2), max(M1, M2))
```

let $T(n)$ denote (worst) case # of comparisons by minmax(A, 1, n).

Then

$$T(n) = \begin{cases} 0 & \text{if } n=1 \\ T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + 2 & \text{if } n \geq 2 \end{cases}$$

By master thm!

simplify : $T(n) = 2T(\frac{n}{2}) + 1$

compare 1 to $n^{\log_2(2)} = \boxed{n^1}$ ← winner

case 3: $\boxed{T(n) = \Theta(n)}$ ← asym. soln.

Exercise show (by direct substitution) that $\boxed{T(n) = 2n-2}$ is exact solution so $T(n) \sim 2n$

Exercise:

Design a recursive algorithm that does exactly $\lceil \frac{3n}{2} \rceil - 2$ comparisons

(Hint: sec 9.1 describes an iterative algorithm that does this.)