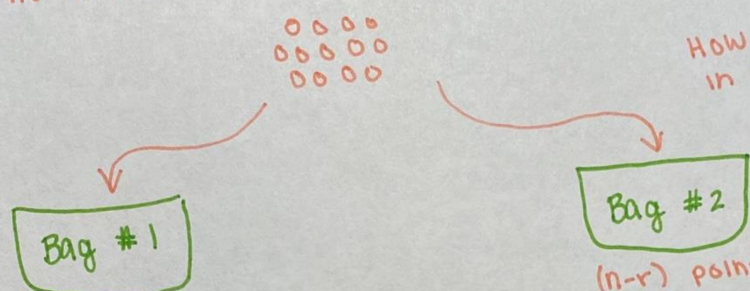
Morgan Baccus
Cpts 350
Homework #6

Problem 1

Find the average case complexity of badClosestPair.

Have n number of points:



How many points in each bag?

r points

(n-r) points

Recursively run the algorithm:

1. split n points between bag 1 and 2.
2. badClosestPair on bag 1 and 2.
3. Conquer part.

$$T_{avg}(n) = \sum_{r=0}^{n} Prob(r)\left[\begin{array}{c}\text{step 1:}\\ \text{let it be}\\ \text{too small}\end{array}\right] + \underbrace{T_{avg}(r) + T_{avg}(n-r)}_{\text{step 2}}$$

$$+ \underbrace{O(r(n-r))}_{\text{step 3}}$$

$Prob(r) = ?$

Assume that P = the probability that a point is thrown into bag #1,

$(1-p)$ = ... bag 2

$Prob(r) = C_n^r \, p^r \, (1-p)^{n-r}$ ← binomial distribution

Problem 1 continued

$$C_n^r = \frac{n!}{r!(n-r)!}$$ ← Bernalli Formula

Now: $$T_{avg}(n) = \sum_{r=0}^{n} \frac{n!}{r!(n-r)!} * \left(\frac{1}{2}\right)^n \left(T_{avg}(r) + T_{avg}(n-r) + O(r(n-r))\right)$$

normal distribution: $$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

↓ Better: ① $\text{Prob}(\mu-\sigma) = \text{Prob}(\mu+\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-1/2}$

② $$\sum_{\mu-\sigma \leq r \leq \mu+\sigma} \text{Prob}(r) \geq 2/3$$

Guess: $T_{avg}(n) = O(n^2 \log n)$

should not attempt to solve as this is unsolvable.

Problem 2

Compute the worst-case complexity of naiveKaratsuba.

$T(n) = T(n-1) \log_2 3$

$T(n-1) = T(n-2) \log_2 3$

Since time complexity here is: $O(n \log_2 3)$

$T(2) = n \log_2 3$

Substitution:

$T(n) = n(\log_2 3)^n$

$T(n) = n^{.59n}$

$\sum_{k=1}^{n-1} O((kn)^{1.59})$

$= O(n^{1.59}) \sum_{k=1}^{n-1} k^{1.59}$

$= O(n^{1.59} \times n^{2.59})$

$\boxed{= O(n^{4.18})}$

Problem 3

Compute worst case complexity of better Karatsuba.

Given: $\alpha_1, \dots, \alpha_n$

$\leftarrow n$ bits $\rightarrow$

first: $\underbrace{\alpha_1, \dots, \alpha_{n/2}}_{x}$, $\underbrace{\alpha^{n/2+}, \dots, \alpha_n}_{y}$

Second: We know Karatsuba over $x-y$

$G(m,n)$ = the worst case time for better Karatsuba over m number of n-bit strings.

Time complexity:

$$G(n,n) = G(n/2, n) + G(n/2, n) + O(n/2, n)^{1.59}$$

We will rename $G(m,n)$ as $F(m)$.

$$F(m) = 2F(\tfrac{n}{2}) + O(\tfrac{n^2}{2})^{1.59}$$

Guess: $F_w(m) = O(n/2 \times n)^{1.59}$

$$O(n^{1.59}) \times \int_0^n x^{1.59}\, dx$$

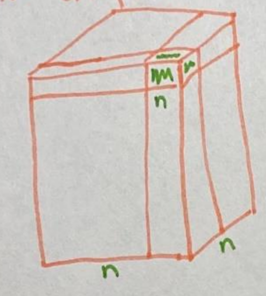
$$= \left[\frac{x^{2.59}}{2.59}\right]_0^n \times an^{1.59}$$

$$= \frac{n^{2.59}}{2.59} \times an^{1.59} \qquad \boxed{= O(\tfrac{n}{2} \times n)^{1.59}}$$

Problem 4

Design an algorithm that runs in time $O(n^3)$ and finds the closest pair of airplanes.

n^3 airplanes   any 2 z

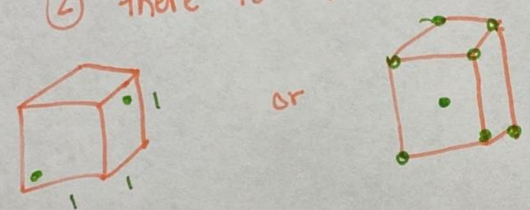


cut n^3 into many unit cubes.

Two cases: ① one unit cube, one airplane.

② there is a unit cube holding > 1 airplane.

Case 2:



Code:

```
Closest Pair (Px, Py) {
If |Px| == 2 : return dot (Px[1], Px[2])
d1 = Closest Pair ( First Half (Px, Py))
d2 = Closest Pair ( second Half (Px, Py))
d = min(d1, d2)

Sy = points in Py within d of L
for i = 1, ...., (sy):
   for j = 1, .... ls :
      d = min (dist (sy[i], sy[j]), d)
return d;
```

Problem 5

Given: α_1 $\#_a(\alpha_1)$, $\#_b(\alpha_2)$

$\vdots$

α_n $\#_a(\alpha_a)$, $\#_b(\alpha_n)$

Want: $O(nm)$ time algorithm to find closest pairs.

Classic closest pair $O(n\log n)$ time so if $\log n > m$, then the classic CP is over the budget of (nm).

We can check if $\log n > m$, then $n > 2^m$, so don't even need to find closest pair.

$\log n > m$

$\uparrow$ # of binary steps. $\alpha_1, \ldots, \alpha_n$

$\uparrow$ length of binary steps

$n > 2^m$

Code:

$O(nm)$

```
If   log n > m
     return 0

else
     run ClosestPair
```

$\boxed{n\log n}$