Refo Yudhanto

CS350

HW 4

1. Formula:
   $T(n)=\Sigma(0<r<n)\ Prob(r) * (T(n-r)+T(r)+O(r(n-r)))$
   Guess:
   $T(n) = O(n^2 logn) = cn^2 logn$
   Sol:
   We divide r into two set B2 $\{r : \mu - \sigma \leq r \leq \mu + \sigma\}$ and B1 is the rest
   $(T(n-r)+T(r)+O(r(n-r)))$ into $X(r)$
   $T(n) = \Sigma r\in B1\ Prob(r) \cdot X(r) + \Sigma r\in B2\ Prob(r) \cdot X(r)$.
   We can conclude $X(\mu - \sigma) \geq 9 \cdot X(1/6 \cdot (\mu - \sigma))$.
   $T(n) \leq 2 \cdot X(1/6 \cdot (\mu - \sigma)) + \Sigma r\in B2\ P\ rob(r) \cdot X(r)$.
   $T(n) \leq (1 + 1/3) \cdot \Sigma r\in B2\ Prob(r) \cdot X(r)$.
   $Prob(r) \leq Prob(\mu)$
   $T(n) \leq (1 + 1/3) \cdot 1 /(\sigma \surd 2\pi) \cdot (2 \cdot \Sigma r\in B2\ T(r) + \Sigma r\in B2\ O(r(n - r)))$.
   $T(n) \leq (1 + 1/3) \cdot 1 /(\sigma \surd 2\pi) \cdot 2 \cdot X\ r\in B2\ T(r) + O(n^2)$.
   Can be counted as
   $T(n) \leq (1 + 1/3) \cdot 1 /(\sigma \surd 2\pi) \cdot 2 \cdot c \cdot (n/2)^3 \cdot \log n/2 \cdot (3 \cdot 2)/\surd n + O(n^2)$.
   $\leq cn^2 logn = O(n^2 logn)$

2. $=\Sigma(n-1)\ (k=1)\ O((kn)\ 1.59)$
   $O(n)^{1.59} *\Sigma(n-1)\ (k=1)(k^{1.59})$
   $O(n^{1.59}*n^{2.59})$
   $=O(n^{4.18})$

3. $G(n) \leq 2 \cdot G(n/2) + b \cdot (n/2 \cdot n)^{1.59}$
   Guess $G(n) = O((n^2)^{1.59}) = c ((n^2)^{1.59})$
   $c>>b$
   $LHS \leq 2 \cdot c \cdot (n/2)^{(2\cdot 1.59)} + b \cdot (n/2 \cdot n)^{1.59} \leq c ((n^2)^{1.59})$
   Thus ,worst complexity is $O((n^2)^{1.59})$.

4. We can split it into n^3 unit cubes and there are $O(n^3)$ eight cubes. There are 2 cases to consider. Case 1 is when there's exactly in each of unit cubes. Thus closest pair cannot be farther than $2\surd 3 < 4$. Case 2 is when there's at least 2 airplanes in some unit cubes, means closest pair cannot be farther than $\surd 3<4$. From this we can conclude that pair cannot be farther than 4. Since there is at least 1 airplane in a half cube, means there are at most 16^3 airplanes in any eight cube. Thus it means it is bounded by $O(n^3)$

5. We can just compute the points in the array A and that will give us O(nm). Since A[i]=A[j] but i/=j, the smallest difference of between the pairs is 0, hence running time is bounded by O(nm).