CSE 326 - Assignment 5

Ahmed Salman Abdelaziz (id: 120180047)

May 3, 2021

Question 1

Claim: The following permutation always results in a $1-(n-2)$ cut: $1, 3, \dots, 2\times\lceil\frac{n}{2}\rceil - 1, 2\times\lfloor\frac{n}{2}\rfloor, 2\times\lfloor\frac{n}{2}\rfloor - 2, \dots, 2$. This is basically the odd numbers sorted then the even numbers inversely sorted.

Proof:

For the first recursive call, the median of the three elements is 2 which results in the following permutation after pivoting: $1, 2, 3, 5, \dots, 2\times\lceil\frac{n}{2}\rceil - 1, 2\times\lfloor\frac{n}{2}\rfloor, \dots, 4$. Notice that the elements starting from the third index can be represented as follows: $1+2, 3+2, \dots, 2\times\lceil\frac{m}{2}\rceil - 1+2, 2\times\lfloor\frac{m}{2}\rfloor + 2, 2\times\lfloor\frac{m}{2}\rfloor - 2+2, \dots, 2+2$ where $m = n-2$. Which is similar to the original permutation but with a smaller n. Hence every recursive call will result in a $1-(n-2)$ cut.

Question 2

The running time is $\Theta(n^2)$, assuming consistency in comparisons (i.e. if the comparison operator is either $\leq$ or $\geq$ not a mix of both). This is because at every recursive call, the pivot will be either placed in the beginning or the end.

Question 3

At every call, the array of size n is split to two with lengths αn and $(1-\alpha)n$. Since $0 < \alpha \leq \frac{1}{2}$, then $(1-\alpha)n \geq \alpha n$. Assuming the level of the maximum depth is k, then we have $(1-\alpha)^k n = c$ for some constant c. Hence $k = \frac{\log c - \log n}{\log 1-\alpha}$. We can simply discard the $\log c$ term as it has no effect on the analysis. Similarly, the level of the minimum depth is $k = \frac{\log c - \log n}{\log \alpha}$.

Question 4

1. The (inversely) sorted permutation will always cause the sorting to happen in $\Theta(n^2)$.

2. Simply shuffle the elements before feeding them to the algorithm. Shuffling by `Fisher-Yates` algorithm runs in $\mathcal{O}(n)$.

Question 5

Let's call a set S of intervals pure if every pair of intervals in this set overlap. Also, we will consider an interval i_k less than an interval i_j if $a_k < a_j$.

Claim 1: Every subset S of (not necessarily sorted) consecutive intervals satisfying that $|S| \leq d$ can be partitioned to at most 2 pure subsets.

Proof:

Assume that this is not true. Then $\exists i_k, i_j, i_l \in S, i_j \cap i_k = i_j \cap i_l = i_k \cap i_l = \phi$. But since $|S| \leq d$ and all intervals in S are consecutive, then the middle interval in $i_j, i_k,$ and i_l can not intersect $d-1$ intervals, which is a contradiction.

Claim 2: If S is a pure set, then any permutation of its elements is a valid, fuzzily-sorted permutation.

Proof:

Let i_j and i_k be the minimal and maximal elements in S, respectively. Hence $\forall i_l \in S, (b_j \leq b_l) \wedge (a_l \leq a_k)$. This means that $\forall i_l \in S, [a_k, b_j] \subset i_l$. Implying that any permutation of S is fuzzily-sorted.

Now, we can make a deterministic quick-sort procedure to divide the intervals into subproblems, until the length of the subproblem is less than d, which will be solved in $\mathcal{O}(d)$ for a total of $\mathcal{O}(n \log \frac{n}{d})$ complexity.

Algorithm 1: Fuzzy-sort n intervals

Input : n intervals
Output: n fuzzy-sorted intervals

Function `SortD(`S`)`**:**
  $mn \leftarrow$ minimum of S
  $mx \leftarrow$ maximum of S
  Let A be an empty doubly linked list
  for $i \leftarrow S$ **do**
    if $i \cap mn \neq \phi$ **then**
      Add i to the beginning of L
    else
      Add i to the end of L
    end
  end
  return L
Function `Main(`S`)`**:**
  if $|S| \leq d$ **then**
    return SortD(S)
  end
  $m \leftarrow$ median of S
  $L, R \leftarrow$ partition of S around m
  $L \leftarrow$ Main(L)
  $R \leftarrow$ Main(R)
  return Concatenation of L, [m], and R