Parth Bakare

Interview 1

There are n people, and their immunities are given in an array of size n. A person dies if his/her immunity is strictly less than the immunity of the person to his/her immediate right. All the people who die as a result are removed from the array (their corresponding immunities are removed from the array), and then the process continues the next day (person i dies if immunity(i) < immunity(i+1)). This process goes on indefinitely.

```
Eg : n = 10
Immunities : 50 35 70 57 41 95 13 7 1 85
People infected on day 1 : 35, 41, 1
New array : 50 70 57 95 13 7 85
People infected on day 2 : 50, 57, 7
New array : 70 95 13 85
People infected on day 3 : 70, 13
New array : 95 85
```

(a) Print the first k people to die in this manner (in the correct order). Create a stack to store indices, such that the immunities represented by these indices form a non increasing sequence. To do this, traverse the array from the left, and before adding the index of the current element, pop out the top index from the stack until the stack is empty or the index at the top corresponds to an immunity greater than or equal to the current immunity. The popped indices are the people who will eventually die. The day on which a person i dies is given by *(index of the first person after i with a greater immunity than i) - i*.

We'll have the day on which every person dies, then sort the people daywise (by index on the same day) and print the first k people (indices).

```
Immunities : 50 35 70 57 41 95 13 7 1 85
stack : []

i = 0, immunity = 50
stack : [0] // after adding 50

i = 1, immunity = 35
stack : [0 1] // after adding 35

i = 2, immunity = 70
stack : [2] /* pop out 0 and 1 since 50 and 35 would die because of 70, 50
dies on day (2 - 0 =) 2
and 35 dies on day (2 - 1 =) 1 */

i = 3, immunity = 57
stack : [2 3]

i = 4, immunity = 41
stack : [2 3 4]
```

```
i = 5, immunity = 95
stack : [5] /* pop out 2, 3 and 4 since 70 57 and 41 would die because of
95, day(2) = 5 - 2 = 3,
day(3) = 5 - 3 = 2 and day(4) = 5 - 4 = 1 */

(continue this process till i < n)
```

(b) Assuming the process goes on until no more people can die, find the number of days it would take for the process to terminate.

Same as (a), number of days = max(day(i))

(c) Assuming the process goes on until no more people can die, rearrange the array so that exactly K people will die.

Sort the array, select the largest k+1 immunities and place them in the beginning in a non-decreasing order, and the remaining n-k-1 people after this is a non-increasing order.

```
Immunties : 50 35 70 57 41 95 13 7 1 85
k = 3
sorted array : 1 7 13 35 41 50 57 70 85 95
rearranged array : 57 70 85 95 50 41 35 13 7 1
```

If some immunties are equal then we may have to select more than k+1 people (select k people with immunties smaller than the largest immunity, place them in the beginning, and then all those with maximum immunity)

(d) Assuming the process goes on until no more people can die, print the daywise list of people that would die.

Same as (a), sort the array daywise.

(e) Assuming the process goes on until no more people can die, rearrange the array so that exactly K people will die and the rearranged array will have minimum edit distance with respect to the original array (edit distance wasn't precisely defined, but try and reduce the difference in indices of every element in the original array and the rearranged array).

The interviewer was only interested in how I would approach this, and I'm not sure if my approach was correct. Basically, we find the number of people that would die, and swap adjacent elements wherever we need to, and every swap would increase or decrease the number of people who die by 1.

Interview 2

A network of computers connected by cables is represented by a graph.

(a) A source computer has a packet which it sends to all its neighbours, who in turn send the packet to all their neighbours. Travelling through a single cable takes 1s. Find the maximum time taken for all computers to receive the packet.

Use BFS.

(b) Multiple source computers are present, and they send out the packet simultaneously. Find the maximum time taken for all computers to receive the packet.

Combine all the sources into a single supernode which is connected to all those nodes which were connected to atleast one source initially. Start BFS from this supernode.

(Insert all the sources in the queue in the beginning, and proceed with the normal BFS).

(c) Single source, travelling through a cable takes either 0s or 1s. Find maximum time.

Use 0-1 BFS.

(d) Single source, cables have positive weights.

Use Dijkstra's theorem

(e)Single source, packet can travel through atmost k cables. Find maximum time.

DP based solution. DP(i)(j) = time taken to reach node i through atmost j edges. DP(i)(0) = INF, if i isn't connected to the source, DP(i)(0) = weight of edge between source and i otherwise.

DP(i)(k) = minimum(dp(i)(k-1), dp(j)(k-1) + weight(j, i)) for all j to which i is connected. k = 1 to k

(f) Single source, graph is divided into connected components. Find the minimum number of cables that have to be replaced (remove an edge and place it somewhere else) so that all computers receive the packet. Print -1 if it is impossible to connect the entire graph.

Using BFS, find every connected component. Number of edges which need to be replaced = Number of connected components - 1 (imagine the components forming a tree). Number of extra edges (which can be replaced) = sum(number of edges in component i - (number of nodes in component i - 1)), since every component needs atleast (size - 1) edges to remain connected. Answer = Number of replacements (or -1 if this number is greater than number of extra edges).

HR/Manager Round

The HR manager wanted a feedback for the entire hiring process, and told me I could ask her any questions if I wanted to. Was more of a chit-chat than an interview.

Tips for the Nutanix Interviews

The questions asked in every round weren't difficult and you'll need to be calm and patient to answer them. If your preparation is thorough, you would have seen similar questions elsewhere. Even the interviewers were very friendly and gave hints if I got stuck somewhere. Regardless of whether you know or don't know how to solve the question, keep discussing your approach with the interviewer(s).

Tejas Khairnar

Debugging Round

Before the online interviews we had a online debugging round wherein we were given a code with some 10-12 logical bugs and we were asked to fix it in a time frame of 45 minutes.

Tips for the Nutanix Interviews

The two Techincal rounds had the same questions for all the students and HR was a chit chat session as Parth mentioned. There is a speculation tha Nutanix considers the scores of all the rounds right from the coding round , debugging round ,.. to the HR round. So it is really important to perform really well in all the rounds and remember each round is an eliminator(not sure about HR).

Here are some stats:

```
Round                   Students shortlisted
Coding round                    35
Debugging round                 14
Round 1                         10
Round 2                         10
```

Nutanix is one of the few comapnies whose interviews are entirely cp based.So it is really important for one to be firm with the basics of competitive coding.