Greedy 2

Wednesday, 23 June 2021 8:03 PM

1) N students, all N students are rated, [Teacher decide to give atleast 1 chocolate every student]

{ if marks[i] > marks[i-1] && choclate[i] > choclate[i-1] || if marks[i] > marks[i+1] && choclate[i] > choclate[i+1] }

→ Help teacher to distribute min number of chocolate as a whole

Ex1:

Marks: 2 4 15 6 17

Choc: 1 2 3 2 3 // C: 11

Choc: 1 2 3 1 2 C: 9

Ex2:

Marks: 3 -1 -4 8

Choc: 3 2 1 2 } C: 8

Ex3:

Marks: 3 5 2 7 0

Choc: 1 2 1 2 1 } C: 7

Ex4:

Marks: 5 7 4 2 0 -1 } C: 11

Choc: 1 5 4 3 2 1

Ex5:

Marks: 5 7 7

Marks: 1 2 2 *

1 2 1 //

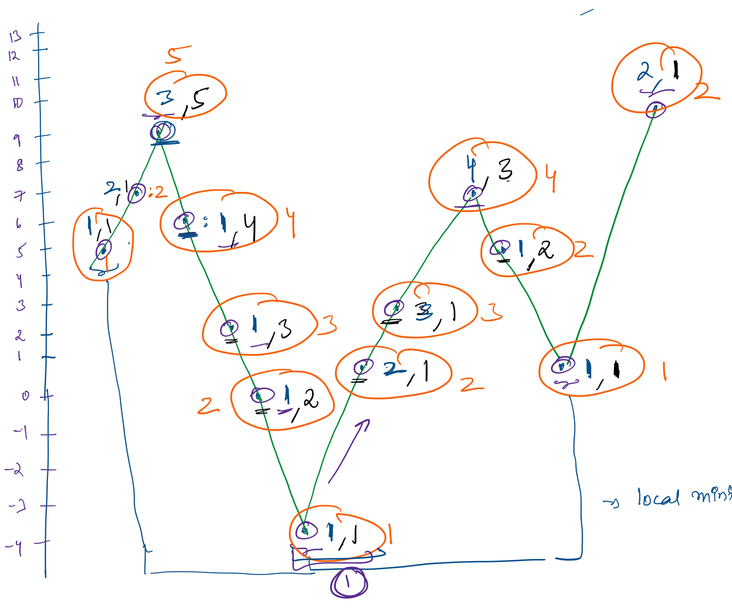
{ if marks[i] > marks[i-1] && choclate[i] > choclate[i-1] } { if marks[i] > marks[i+1] && choclate[i] > choclate[i+1] }

Ex6: [5 7 9 6 2 0 -4 1 3 7 5 1 10]

Left Ch: (L-R)	1	2	3	1	1	1	1	2	3	4	1	1	2
Right Ch: (R-L)	1	1	5	4	3	2	1	1	1	3	2	1	1
Max(Left & Right):	1	2	5	4	3	2	1	2	3	4	2	1	2

Pseudocode

1) Construct cho1[] with only left condition
2) Construct cho2[] with only right condition
3) for i = 0 to N-1: cho3[i] = max(cho1[i], cho2[i])



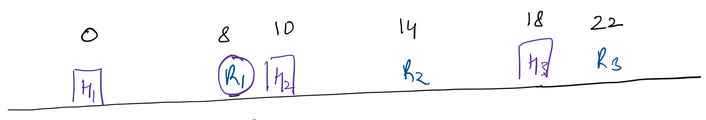
Greedy Steps:

1) local minima = 1
2) ar[i] > ar[i+1]

cho[i] >= cho[i+1] + 1

2) Gaurav: 3 rats, 3 holes]) In a hole can have 1 rat

2) Rat speed 1km/1hr, All started to escape at same time. Find minimum time such all of them can reach a safe house



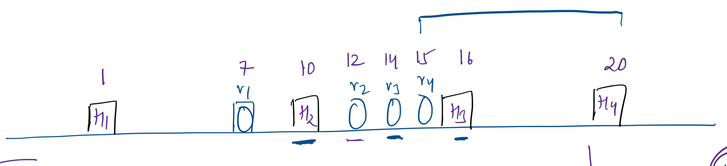
R1 — H2 : 2
R2 — H1 : 14
R3 — H3 : 4
} 14 hrs

R1 — H1 : 8
R2 — H2 : 4
R3 — H3 : 4
} 8

Max of all

→ Q) Given N rats & N holes, find min time so that all rats can reach a holes.

Ex:

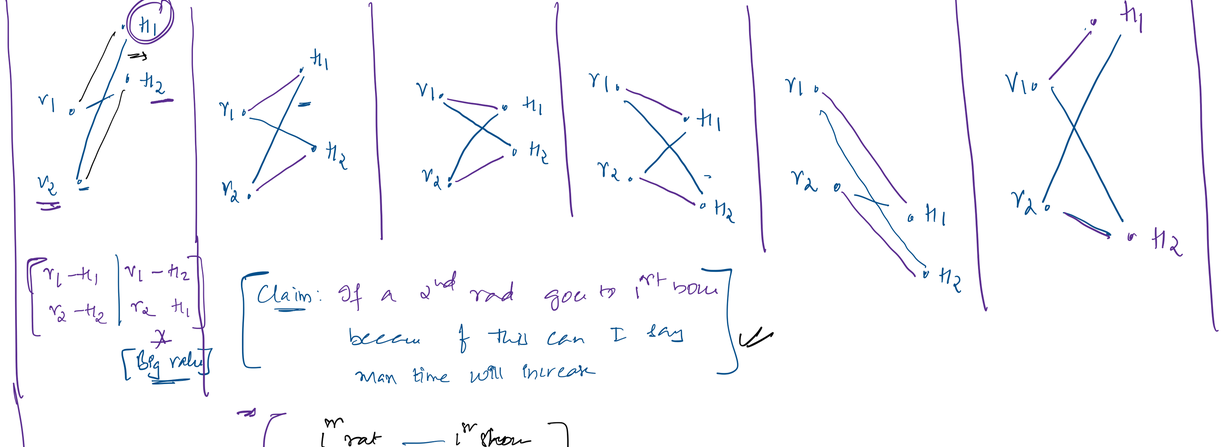


Solution:

1) Sort all H[]
2) Sort r[]

dist = 0
for (int i = 0; i < N; i++)
  dist = max(dist, abs(r[i] - H[i]))
return dist;

1 H1 — 7 r1
10 H2 — 12 r2
16 H3 — 14 r3
20 H4 — 15 r4



File: A B A C B D A E A F A B B C D E F A F A E ...

{ A - 50, C - 20, E - 10, B - 30, D - 15, F - 5 }

Total character = 130 characters

1 single character = 1 Byte

Total size = 130 Bytes → 130 Bytes → 57 Bytes

Fixed length encoding

0 0 0	A
0 0 1	B
0 1 0	C
0 1 1	D
1 0 0	E
1 0 1	F

Each character = 3 bits

Total character = 130 *

Total size = 130 * 3 = 390 bits

390/8 = 49 Bytes

To Decode store table as well [Table size]

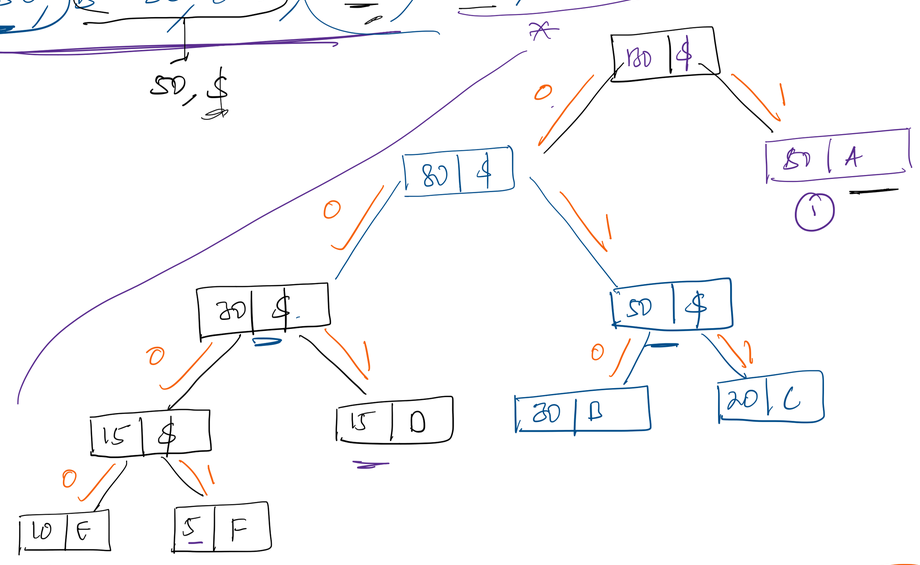
⇒ 18 bits + 6 * 8 : 66 bits

Total size:
390
66
456
456 bits
57 bytes

Issues:

	Bits used
16 distinct characters →	4 bits
2^5 distinct characters →	5 bits
2^7 distinct characters →	7 bits
2^8 distinct character →	8 bits → 1 Byte

A - 50, B - 30, C - 20, D - 15, E - 10, F - 5



			bits
F	: 0000	: 10	40
E	: 0001	: 5	20
D	: 001	: 15	45
B	: 010	: 30	90
C	: 011	: 20	60
A	: 1	: 50	50
			305

48 bits + 18 bits

66

371

→ higher freque lesser then encoding

Data

1) Frequency table
2) Huffman Tree
3) [Encoding Table]

* We no longer need frequency table or huffman tree