

Date: 1402-02-07

تمرین سری دو طراحی الگوریتم - بهار ۱۴۰۲

Email: Rahmat2022a@gmail.com

لینک سوالات و جوابها: github.com/EnAnsari/algorithm-hsu

$$W = 30$$

$$w_i = \{5, 10, 20\}$$

$$p_i = \{50, 60, 140\}$$

$$p_i/w_i = \{ \frac{50}{5} = 10, \frac{140}{20} = 7, \frac{60}{10} = 6 \}$$

(۱) ابتدا بر اساس p_i/w_i مرتب می‌کنیم. s_1 s_2 s_3

حل: s_1 و وزن باقی‌مانده 30 است می‌توان تمام می‌شود. اول را برداشت

$$Ans = \{s_1\}$$

$$w = 5$$

$$p = 50$$

سپس می‌بینیم که

وزن باقی‌مانده 25 است و باقی‌مانده 25 است و می‌توان در آن را به مرتبه سوره که وزن آن 20 است را هم می‌توان برداشت.

$$Ans = \{s_1, s_3\}$$

$$w = 20$$

$$p = 140$$

وزن باقی‌مانده 5 است و باقی‌مانده 5 و می‌توان

می‌توان به دست آورد (الگوریتم)

$$total\ p = 140 + 50 = 190$$

تمام است.

character	a	b	c	d	e	f	g	h
Frequency	1	1	2	3	5	8	13	21
code								

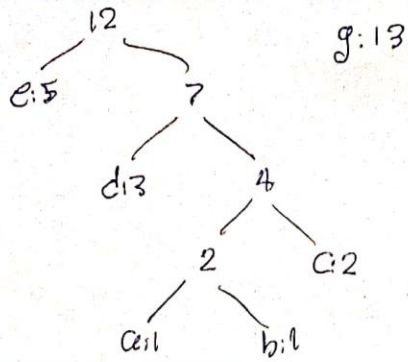
step 1: a:1 b:1 c:2 d:3 e:5 f:8 g:13 h:21

step 2: a:1 b:1 c:2 d:3 e:5 f:8 g:13 h:21

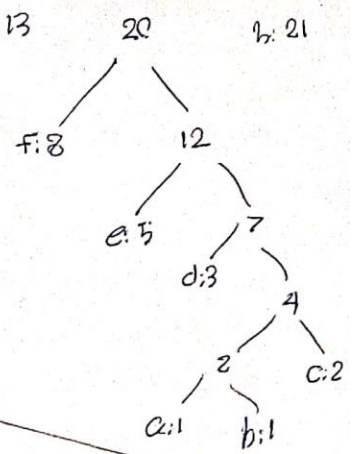
step 3: a:1 b:1 c:2 d:3 e:5 f:8 g:13 h:21

step 4: a:1 b:1 c:2 d:3 e:5 f:8 g:13 h:21

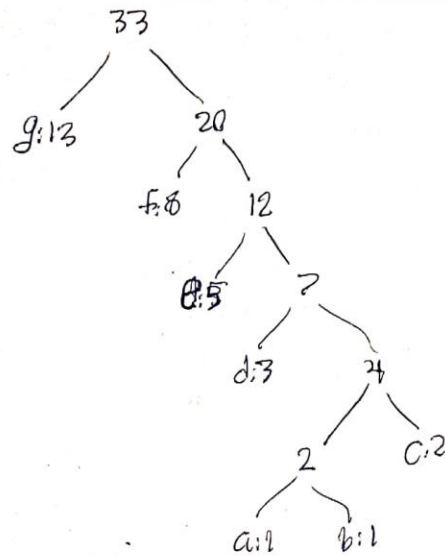
step 5: f: 8



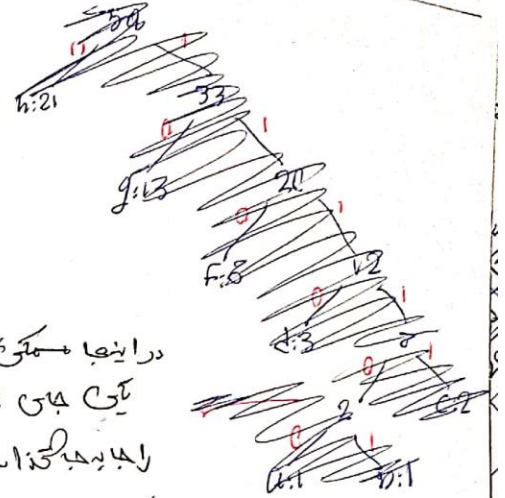
step 6: g:13



step 7: h:21



finest step



در اینجا ممکن است
یک 4 یا 2، 2
را به جای 2، 2
که جواب متفاوت
ایجاد خواهد کرد اما
باز هم درست است.

h → 0

g → 10

f → 110

~~d → 1110~~

e → 11110

d → 11110

c → 11111

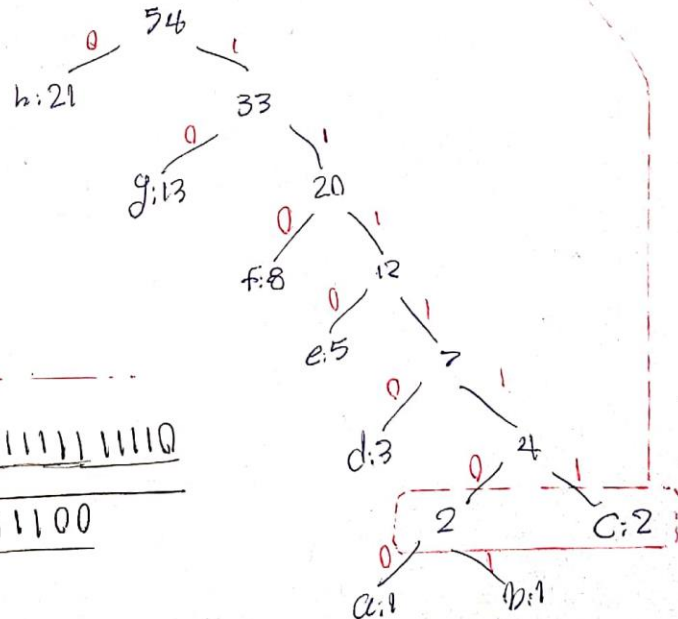
a → 1111100

b → 1111101

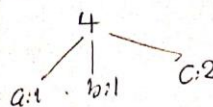
Final
Answer

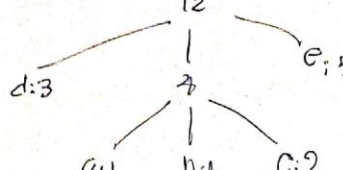
aaghecd = 1111100 1111100 100 1110 11111 11110

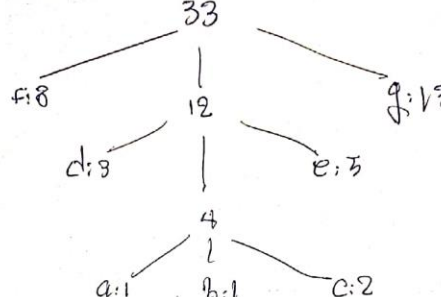
fgdea = 110 10 11110 1110 1111100



step 1: a:1 b:1 c:2 d:3 e:5 f:8 g:13 h:21 1/38

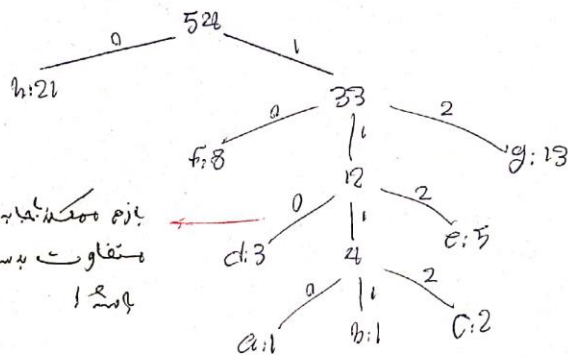
step 2: d:3  e:5 f:8 g:13 h:21

step 3: f:8  g:13 h:21

step 4: h:21 

final step

این مسئله را می توان به صورت یک مسئله دینامیک برنامه نویسی در نظر گرفت



h: 0
f: 10
g: 12
d: 110
e: 112
a: 1110
b: 1111
c: 1112

final answer

uaghecd: 111011101201121112110

fgdeca: 10121101121110

$$m_{i,j} = \begin{cases} 0 & i=j \\ \min_{i \leq k < j} \{ m_{i,k} + m_{k+1,j} + d_{i-1} \times d_k \times d_j \} & i < j \end{cases}$$

$i=j$

$i < j$

1/38

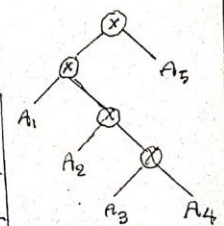
$d_0 = 10$
 $d_1 = 4$
 $d_2 = 5$
 $d_3 = 20$
 $d_4 = 2$
 $d_5 = 50$

	1	2	3	4	5
1	0	200	1200	320	1320
2	0	0	400	240	640
3	0	0	0	200	500
4	0	0	0	0	2000
5	0	0	0	0	0

m

	1	3	3	4	5
1	0	1	1	1	24
2	0	0	2	2	4
3	0	0	0	3	4
4	0	0	0	0	4
5	0	0	0	0	0

K



این مسئله را می توان به صورت یک مسئله دینامیک برنامه نویسی در نظر گرفت

$$(A_1 \times (A_2 \times (A_3 \times A_4))) \times A_5 \quad \leftarrow \text{Final Answer}$$

Integers	1	2	3	4	5	6
probability	0.01	0.02	0.04	0.08	0.16	0.69

2/63

$$C_{i,j} = \begin{cases} 0 & \text{if } i > j, \\ P_i & \text{if } i = j, \\ \min_{i \leq k \leq j} \{ C_{i,k-1} + C_{k+1,j} + P_{i,j} \} & \text{if } i < j \end{cases}$$

$$P_{i,j} = \sum_{t=i}^j P_t$$

	1	2	3	4	5	6
1	0.01	0.02	0.11	0.26	0.57	1.57
2	0	0.02	0.08	0.22	0.52	1.51
3	0	0	0.04	0.16	0.44	1.41
4	0	0	0	0.08	0.32	1.25
5	0	0	0	0	0.16	1.01
6	0	0	0	0	0	0.69

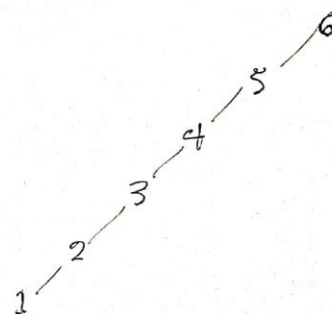
C

	1	2	3	4	5	6
1	0	2	3	4	5	6
2	0	0	3	4	5	6
3	0	0	0	3	5	6
4	0	0	0	0	5	6
5	0	0	0	0	0	6
6	0	0	0	0	0	0

K

Cost of Optimal BST is 1.57

$$C_{xg} = \begin{cases} k=3 & 0 + 1.25 + 0.97 = 2.22 \\ k=4 & 0.04 + 1.01 + 0.97 = 2.02 \\ k=5 & 0.16 + 0.69 + 0.97 = 1.82 \\ k=6 & 0.44 + 0 + 0.97 = 1.42 \checkmark \end{cases}$$



جواب سوال زیر علاوه بر صفحات بعدی در [این لینک](#) + زیر نیز موجود است.

مرتبه زمانی این الگوریتم $O(n \log n)$ و پیچیدگی حافظه آن نیز $O(n)$ می باشد.

انتخاب الگوریتم مرتب سازی در این سوال کاملاً اختیاری است و من برای اینکه مرتبه زمانی خوبی داشته باشیم از الگوریتم merge sort استفاده کرده ام.

```

#include <iostream>

using namespace std;

void merge(int array[], int name[], int const left, int const mid, int const
right) {
    auto const subArrayOne = mid - left + 1;
    auto const subArrayTwo = right - mid;
    auto *leftArray = new int[subArrayOne], *rightArray = new int[subArrayTwo];
    auto *leftName = new int[subArrayOne], *rightName = new int[subArrayTwo];
    for (auto i = 0; i < subArrayOne; i++) leftArray[i] = array[left + i],
leftName[i] = name[left + i];
    for (auto j = 0; j < subArrayTwo; j++) rightArray[j] = array[mid + 1 + j],
rightName[j] = name[mid + 1 + j];
    auto indexOfSubArrayOne = 0, indexOfSubArrayTwo = 0;
    int indexOfMergedArray = left;
    while (indexOfSubArrayOne < subArrayOne && indexOfSubArrayTwo < subArrayTwo)
    {
        if (leftArray[indexOfSubArrayOne] <= rightArray[indexOfSubArrayTwo]) {
            array[indexOfMergedArray] = leftArray[indexOfSubArrayOne];
            name[indexOfMergedArray] = leftName[indexOfSubArrayOne];
            indexOfSubArrayOne++;
        }
        else {
            array[indexOfMergedArray] = rightArray[indexOfSubArrayTwo];
            name[indexOfMergedArray] = rightName[indexOfSubArrayTwo];
            indexOfSubArrayTwo++;
        }
        indexOfMergedArray++;
    }
    while (indexOfSubArrayOne < subArrayOne) {
        array[indexOfMergedArray] = leftArray[indexOfSubArrayOne];
        name[indexOfMergedArray] = leftName[indexOfSubArrayOne];
        indexOfSubArrayOne++;
        indexOfMergedArray++;
    }
    while (indexOfSubArrayTwo < subArrayTwo) {
        array[indexOfMergedArray] = rightArray[indexOfSubArrayTwo];
        name[indexOfMergedArray] = rightName[indexOfSubArrayTwo];
        indexOfSubArrayTwo++;
        indexOfMergedArray++;
    }
    delete[] leftArray;
    delete[] rightArray;
    delete[] rightName;
}

```

```

        delete[] leftName;
    }

void mergeSort(int array[], int name[], int const begin, int const end) {
    if (begin >= end) return;
    auto mid = begin + (end - begin) / 2;
    mergeSort(array, name, begin, mid);
    mergeSort(array, name, mid + 1, end);
    merge(array, name, begin, mid, end);
}

int main() {

    int n, temp;
    cout << "Enter number of persons: ";
    cin >> n;

    int pay[n] = {0}, name[n];

    for(int i = 0; i < n; i++) {
        name[i] = i;
        for(int j = 0; j < n; j++) {
            if(i == j) continue;
            cout << "p" << i << " has to pay ? Rs to p" << j << " : ";
            cin >> temp;
            pay[i] -= temp;
            pay[j] += temp;
        }
    }

    mergeSort(pay, name, 0, n - 1);

    for(int i = 0, j = n - 1; pay[i] < 0;) {
        if(pay[i] * (-1) < pay[j]) {
            cout << "P" << name[i] << " pays " << pay[i] * (-1) << " to Rs to P"
<< name[j] << endl;
            pay[j] -= pay[i] * (-1);
            i++;
        }
        else {
            cout << "P" << name[i] << " pays " << pay[j] << " to Rs to P" <<
name[j] << endl;
            pay[i] += pay[j];
            j--;
        }
    }
}

```

```
    }  
    return 0;  
}
```