Nama : Fajar Satria

NRP : 05111940000083

Kelas : B

Resume Kuliah PAA – 19 April 2021

Divide and Conquer Approach

breaks original problem into several smaller instances of the same problem. Salah satu penggunaan divide and conquer approach adalah genetic algorithm karena setiap saat selalu mempertimbangkan adanya kromosom yang lebih fit dengan membandingkan nilai fitness.

Ciri divide and conquer adalah ketika mencari solusi sebuah persoalan original melewati persoalan subproblemnya. Subproblem adalah problem yang memiliki identitas yang sama dengan original problem akan tetapi *instances*-nya lebih kecil.

Sesuatu yang rekursif pasti memiliki sifat rekurens, akan tetapi relasi rekurens belum tentu harus diselesaikan dengan rekursif. Sebagai contoh dalam mencari bilangan Fibonacci.

$$f(n) = f(n-1) + f(n-2)$$

Terlihat adanya pembanggilan fungsi f(n-1) dan f(n-2) yang merupakan subproblem dari f(n). Biasanya problem seperti ini diselesaikan dengan rekursif. Akan tetapi dalam kasus ini dapat diselesaikan dengan non-rekursif. Jadi cukup melakukan perulangan sampai batas tertentu dan menyimpan nilai yang dihasilkan sebelumnya, setelah mencapai batas bisa didapatkan hasilnya.

a. Recursive Function 1

$$f(n) = f\left(\frac{n}{2}\right) + f(\frac{n}{3})$$

Persoalan ini terlihat mudah. Akan tetapi yang menjadi masalah adalah nilai **n** memiliki batas sampai **10^18** dan dengan batas eksekusi **1** detik.

Solusi yang paling mungkin adalah dengan pendekatan divide and conquer dan dengan fungsi rekursif dan dengan menyimpan nilai yang sudah dihitung dalam struktur data map. Hal ini dilakukan karena pada kasus tertentu $f(\frac{n}{2})$ dan $f(\frac{n}{3})$ akan memiliki nilai yang sama. Oleh karena itu dalam fungsi rekursif tersebut cukup diberi batasan baru yakni ketika sudah pernah menghitung suatu nilai cukup me-*return*-kan hasil yang sudah disimpan dalam map tersebut.

PSEUDOCODE

REC(u)

1. Let mp new map
2. mp[0] = 1
3. if exists mp[u]
4. return mp[u]
5. else mp[u] = REC(u/2) + REC(u/3)
6. return mp[u]

SOURCECODE

```
#include<map>
#include<cstdio>
using namespace std;
typedef long long ll;
map<ll,ll> mp;

ll rec(ll u){
    if(mp.count(u)){
```

```
        return mp[u];
    }
    ll nval = rec(u/2) + rec(u/3);
    mp[u] = nval;
    return nval;
}

int main(){
    mp[0] = 1;
    ll n;
    scanf("%lld",&n);
    printf("%lld\n",rec(n));
    return 0;
}
```

SUBMISSION

#	Submit date	Lang	Time	CPU	Memory	State
8873336	Apr 19, 2021, 10:00:48 AM	C++ 11 (gnu 10.2)	1 ms	1 ms	1746	✓ Accepted

b. Wires

Given **N** segments of wire length**L**1, **L**2, ..., **L**Ncentimeters. Required by cutting them to get out of **K** equal intervals as may be of greater length, which is expressed by a number of centimeters. If you can not get even the **K** segments of length **1** cm, output **0**.

Pada persoalan ini dapat diselesaikan dengan divide-and-conquer dengan menggunakan algoritma Binary Search. Binary Search tersebut harus memiliki batas bawah dan batas atas yang benar pula.

Sebelum masuk ke penggunaan Binary Search, perlu dilakukan pencarian panjang total dari seluruh segment dan juga mencari segment terpanjang. Setelah didapatkan kedua nilai tersebut, persoalan ini dapat dibagi menjadi 3 case.

Case pertama, ketika panjang total seluruh segment bernilai kurang dari nilai k maka sudah pasti tidak mungkin diselesaikan. Jadi hasilnya adalah 0.

Case kedua, ketika panjang total seluruh segment bernilai sama dengan nilai k maka sudah pasti hasilnya adalah 1.

Case ketiga, ketika panjang total seluruh segment bernilai lebih dari nilai k maka masuk pada penggunaan binary search. Dengan batas bawah sama dengan 1, dan batasnya adalah nilai paling kecil diantara segment terpanjang ditambah satu dan panjang total seluruh segment dibagi k ditambah 1. Dengan Batasan tersebut akan didapatkan hasil yang valid.

Setelah selesai dengan metode diatas solusi yang dihasilkan masih bisa dimaksimalkan lagi dengan pemanfaatan *fast I/O* dan juga penggunaan *binary shifts.*

PSEUDOCODE

```
CALC(u, n, data)
1. temp = 0
2. for i=0 to n-1
3.       temp = temp + data[i]/u
4. return temp

SOLVE(n, k, data)
1. sum = 0
2. MX = 0
3. for i=0 to n-1
4.       if MX < data[i]
5.                MX = data[i]
```

```
6.      sum = sum + data[i]
7.  If sum < k
8.      return 0
9.  else if sum == k
10.     return 1
11. else
12.     atas = min(MX+1, sum/k+1)
13.     bawah = 1
14.     while (atas-1 != bawah)
15.             mid = (atas+bawah)/2
16.             if CALC(mid)<k
17.                     atas = mid
18.             else
19.                     bawah = mid
20.     return bawah
```

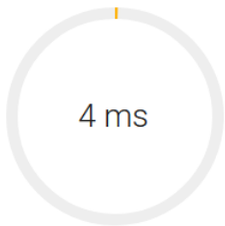
SOURCECODE

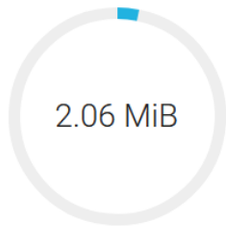
```cpp
#include <cstdio>
#include <iostream>
#include <algorithm>
using namespace std;
typedef long long ll;
int n,k;
ll datas[10010],mid,atas,bawah,sum,MX;

template<typename T>
T getNum(){
    T res=0;
    char c;
    while(1){
        c = getchar();
        if(c==' '||c=='\n') continue;
        else break;
    }
    res=c-'0';
    while(1){
        c = getchar();
        if(c>='0' && c<='9') res=10*res+c-'0';
        else break;
    }
    return res;
}

ll calc(ll u){
    ll temp = 0;
    for(int i=0;i<n;i++) temp += (datas[i]/u);
    return temp;
}
int main(){
    n = getNum<int>();
    k = getNum<int>();
    sum = 0; MX = 0;
    for (int i=0; i<n; i++){
        datas[i] = getNum<ll>();
        if(MX<datas[i]) MX = datas[i];
        sum+=datas[i];
    }
    if(sum<k) printf("0");
    else if(sum == k) printf("1");
    else{
        atas = min(MX+1,sum/k+1);
        bawah= 1;
        while(atas-1 != bawah){
            mid = (atas+bawah) >> 1;
```

```
                if(calc(mid)<k){
                    atas = mid;
                }else{
                    bawah = mid;
                }
            }
            printf("%lld",bawah);
        }
        printf("\n");
        return 0;
}
```

SUBMISSION

Problem	Submitted	Programming Language	Author
Wires	5 hours ago	C++ 11 (gnu 10.2)	Fajar_Satria_05111940000083





#	Submit date	Lang	Time	CPU	Memory	State
8874133	Apr 19, 2021, 11:17:12 AM	C++ 11 (gnu 10.2)	2 ms	2 ms	2056	✓ Accepted
8873921	Apr 19, 2021, 10:59:11 AM	C++ 11 (gnu 10.2)	3 ms	3 ms	2077	✓ Accepted