

Задача о рюкзаке

ты приходишь грабить банк,
а в банке n золотых
слитков, причем про каждый
слиток известно 2 параметра
(масса и стоимость)
 $w[i]$ - weight вес
 $v[i]$ - value стоимость

грузоподъемность $\leq A$ кг

Задача унести максимальную
стоимость в своем рюкзаке

жадным алгоритмом -
пытаешься брать слиток
максимальной стоимости,
но это ошибочная стратегия
 $A=6$

$v[1]=3$ $w[1]=3$

$v[2]=3$ $w[2]=3$

$v[3]=4$ $w[3]=4$

пусть ты i -ый слиток не берешь, тогда
 $a[i][j]=a[i-1][j]$
пусть ты i -ый слиток берешь, тогда
 $a[i][j]=a[i-1][j-w[i]] + v[i]$

1. Какие значения мы вычисляем $a[i]$ -
максимальная стоимость, которую можно
унести проанализировав i первых слитков
2. Какое рекурсивное соотношение
 $a[i]=$
3. Какие начальные значения
4. В каком порядке вычисляются значения
5. Где искать ответ $a[n]$

1. Какие значения мы вычисляем $a[i][j]$ -
максимальная стоимость, которую можно
унести проанализировав i первых слитков,
если твоя грузоподъемность $\leq j$
2. Какое рекурсивное соотношение
 $a[i][j]=\max(a[i-1][j], a[i-1][j-w[i]]+v[i])$
3. Какие начальные значения
 $a[i][0]=0$
 $a[0][j]=0$
4. В каком порядке вычисляются значения
 i from 1 to n
 j from 1 to A
5. Где искать ответ $a[n][A]$
6. from

```
void bag(int n, int A)
{
    int w[n+1]={},v[n+1]={};
    int **a=new int* [n+1];
    for(int i=0;i<=n;i++)
    {
        a[i]=new int [A+1];
    }
    struct trace **from=new struct trace* [n+1];
    for(int i=0;i<=n;i++)
    {
        from[i]=new struct trace [A+1];
    }
    mat_zero_trace(from,n+1,A+1);
    weight_value(w,v,n,A);
    for(int i=0;i<=n;i++)
    {
        a[i][0]=0;
    }
    for(int j=0;j<=A;j++)
    {
        a[0][j]=0;
    }
    for(int i=1;i<=n;i++)
    {
        for(int j=1;j<=A;j++)
        {
            if(j-w[i]>=0)
            {
                a[i][j]=max(a[i-1][j],a[i-1][j-w[i]]+v[i]);
                if(a[i-1][j]>a[i-1][j-w[i]]+v[i])
                {
                    (from[i][j]).x=i-1;
                    (from[i][j]).y=j;
                    (from[i][j]).bullion=0;
                    (from[i][j]).value=0;
                }
            }
            else
            {
                (from[i][j]).x=i-1;
                (from[i][j]).y=j-w[i];
                (from[i][j]).bullion=i;
                (from[i][j]).value=v[i];
            }
        }
        else
        {
            a[i][j]=a[i-1][j];
            (from[i][j]).x=i-1;
            (from[i][j]).y=j;
            (from[i][j]).bullion=0;
            (from[i][j]).value=0;
        }
    }
    print_m(w,n+1);
    print_m(v,n+1);
    print_mat(a,n+1,A+1);
    //print_total_trace(from,n,A);
    print_trace(from,n,A);
    cout<<"a[n][A]= "<<a[n][A]<<endl;
}
```

```
struct trace{
    int x;
    int y;
    int bullion;
    int value;
};
void print_m(int mass[], int length)
{
    for(int i=0;i<length;i++)
    {
        cout<<mass[i]<<" ";
    }
    cout<<endl;
}
void print_mat(int **mass, int y, int x)
{
    //cout.setf(ios::);
    //cout.width(5);
    for(int i=0;i<y;i++)
    {
        for(int j=0;j<x;j++)
        {
            cout<<setw(3)<<mass[i][j]<<" ";
        }
        cout<<endl;
    }
    cout<<endl;
}
void print_trace(struct trace **from, int i, int j)
{
    cout<<(from[i][j]).bullion<<" "<<(from[i][j]).value<<endl;
    if((from[i][j]).x>0 && (from[i][j]).y>0)
    {
        print_trace(from,(from[i][j]).x,(from[i][j]).y);
    }
}
void print_total_trace(struct trace **from, int y, int x)
{
    //cout.setf(ios::);
    //cout.width(5);
    for(int i=0;i<y;i++)
    {
        for(int j=0;j<x;j++)
        {
            cout<<(from[i][j]).bullion<<" "<<(from[i][j]).value<<" "<<(from[i][j]).x<<" "<<(from[i][j]).y<<" ";
        }
        cout<<endl;
    }
    cout<<endl;
}
void mat_zero_trace(struct trace **from, int y, int x)
{
    for(int i=0;i<y;i++)
    {
        for(int j=0;j<x;j++)
        {
            mass[i][j]=0;
        }
    }
}
void mat_zero_trace(struct trace **from, int y, int x)
{
    for(int i=0;i<y;i++)
    {
        for(int j=0;j<x;j++)
        {
            (from[i][j]).bullion=0;
            (from[i][j]).value=0;
            (from[i][j]).x=0;
            (from[i][j]).y=0;
        }
    }
}
void weight_value(int w[], int v[], int n, int A)
{
    for(int i=1;i<=n;i++)
    {
        w[i]=rand()%A+1;
        v[i]=rand()%100+1;
    }
}
```