Shopping with coupons

孙若愚 李经雄 李秋宇

Chapter 1: Introduction

This report introduces an optimized shopping discount algorithm based on a priority queue, which maximizes the number of items purchased within a limited budget. By intelligently matching product prices with coupons, the algorithm ensures that each purchase is made with the greatest possible discount.

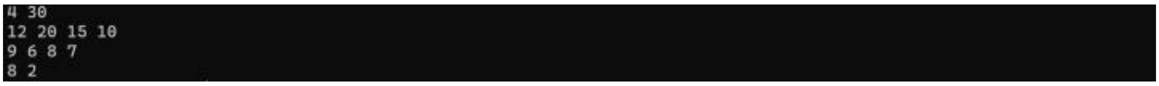
Chapter 2: Data Structure / Algorithm Specification

This section will detail the workings of the algorithm. First, the product prices are sorted in ascending order, and the coupons are sorted in descending order. Then, a priority queue is used to manage all possible combinations of products and coupons, where each node in the queue represents a combination, and its priority is determined by the actual cost of the combination. The pseudocode for the algorithm is as follows:

```
1. Input the number of items N and the budget D
2. Input all item prices and coupon values
3. Sort item prices in ascending order, sort coupons in descending order
4. Initialize priority queue Q
5. For each item i, add (i, 0) to Q
6. While D > 0 and Q is not empty:
    a. Extract the combination with the lowest cost from Q
    b. If D >= cost of the combination:
        i. Subtract the cost from D
        ii. Increment the purchase count
        iii. If there are remaining coupons, add (i, j+1) to Q
    c. Otherwise, exit the loop
7. Output the purchase count and the remaining budget
```

Chapter 3: Testing Results

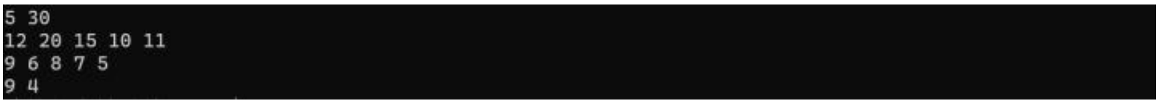
Case 1:



```
4 30
12 20 15 10
9 6 8 7
8 2
```

Passed

Case 2:



```
5 30
12 20 15 10 11
9 6 8 7 5
9 4
```

10*4-9-8-7-6=10

11*3-9-8-7=9

12*2-9-8=7

4+3+2=9,30-10-9-7=4

Passed

Chapter 4: Analysis and Comments

This section will analyze the time complexity of the algorithm as O(NlogN)and the space complexity as O(N)

Appendix: Source Code (if required)

```
#include<cstdio>
#include<queue>
#include<algorithm>
using namespace std;

const int maxn = 100000;
int item[maxn];
int coupon[maxn];
//Sorting function, arranged in descending order
bool cmp(int a, int b)
{
    return a>b;
}
struct node{
    int i,j;
    friend bool operator < (node a, node b){
        return item[a.i]-coupon[a.j]>item[b.i]-coupon[b.j];
    }//overloading the less-than operator
};
int main()
{
    int N,D;
    //input
    scanf("%d%d",&N,&D);
    for(int i=0; i<N; i++){
        scanf("%d",&item[i]);
    }
    for(int i=0; i<N; i++){
        scanf("%d",&coupon[i]);
    }
    //Sort the product prices in ascending order.
    sort(item,item+N);
    //Sort the coupons in descending order.
    sort(coupon,coupon+N,cmp);
    priority_queue<node> q;
    node tmp;
    //Initialize the priority queue
    for(int i=0; i<N; i++){
        tmp.i= i;
        tmp.j = 0;
        q.push(tmp);
    }
    int cost,cnt;
    cnt = 0;
    //Purchasing process
    do{
        tmp = q.top();//take out the next least costly item
        q.pop();
        cost = item[tmp.i]-coupon[tmp.j];
        if(D>=cost){//If the remaining money is more than the expense, then make the purchase.
            D-=cost;
            cnt++;
            if(tmp.j!=N-1){//If it is not the last coupon,
                tmp.j++;//Each coupon can only be used once for each type of product.
                q.push(tmp);
            }
        }
        else break;//Exit when our money is not enough to pay for the minimum amount.
    }while(!q.empty()&&D>0);
    printf("%d %d\n",cnt,D);
    return 0;
}
```

Author List

孙若愚 李经雄   李秋宇

Declaration

We hereby declare that all the work done in this project titled "Shopping With Coupons" is of our independent effort as a group.