作业题目

**动态规划**

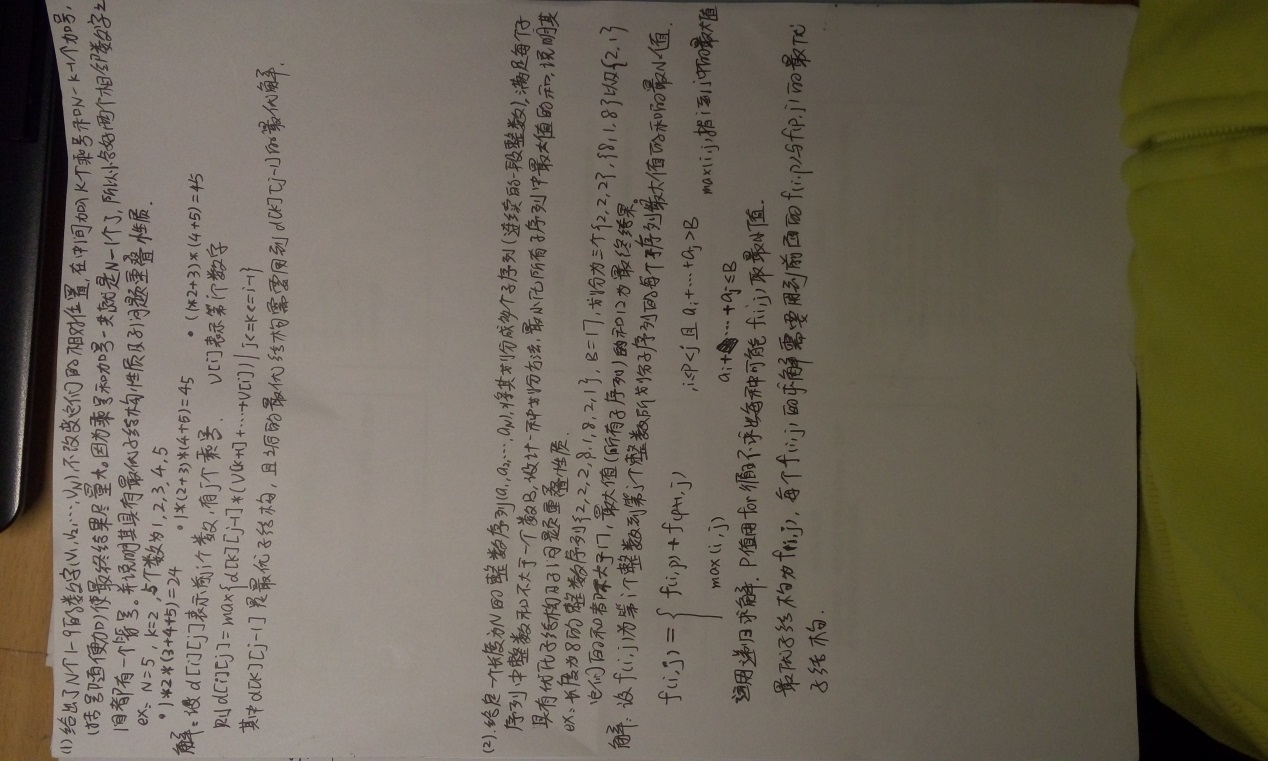
1、给出N个数字，不改变它们的相对位置，在中间加入K个乘号和N-K-1个加号，（括号随便加）使最终结果尽量大。因为乘号和加号一共就是N-1个了，所以恰好每两个相邻数字之间都有一个符号。

**分析**

由于括号可以随意添加，那么最直接的想法是把一段式子分成左右两段，由这两段递推得解。由于不等式的性质，左右当左右两段取到最大值时合起来一定最优（a>c>0, b>d>0⇒a+b>c+d, a\*b>c\*d）。

设F(l, r, k)为[l, r]上数字添加k个乘号算得的最大值，则方程：

F(l, r, k) = A[l], l=r;

****F(l, r, k) = max{ max{F(l, d, k’) + F(d+1, k-k’)} (l≤d<r, 0≤k’≤k, k’≤d-l, k-k’≤r-d-1), max{F(l, d, k’) + F(d+1, k-k’-1)} (l≤d<r, 0≤k’≤k, k’≤d-l, k-k’≤r-d) }

当然，这样的话枚举乘号个数的时候会比较复杂，可以简化。

设F(i, j)为前i个数添加j个乘号的最大值，则方程：

F(i, j) = max{F(i-1, j), F(k, j-1) \* (S[i] - S[k] + A[k])} (0<k<i)

1 #include <iostream>

2 using namespace std;

3 unsigned long long F[102][102];

4 unsigned long long S[102];

5 unsigned long long A[102];

6 int N, K;

7 int main()

8 {

9 cin >> N >> K;

10 for (int i = 1; i <= N; ++i) {

11 cin >> A[i];

12 S[i] = S[i - 1] + A[i];

13 }

14 for (int i = 1; i <= N; ++i) {

15 F[i][0] = F[i - 1][0] + A[i];

16 for (int j = 1; j < i && j <= K; ++j)

17 for (int k = 1; k != i; ++k)

18 if (F[k][j - 1] \* (S[i] - S[k]) > F[i][j])

19 F[i][j] = F[k][j - 1] \* (S[i] - S[k]);

20 }

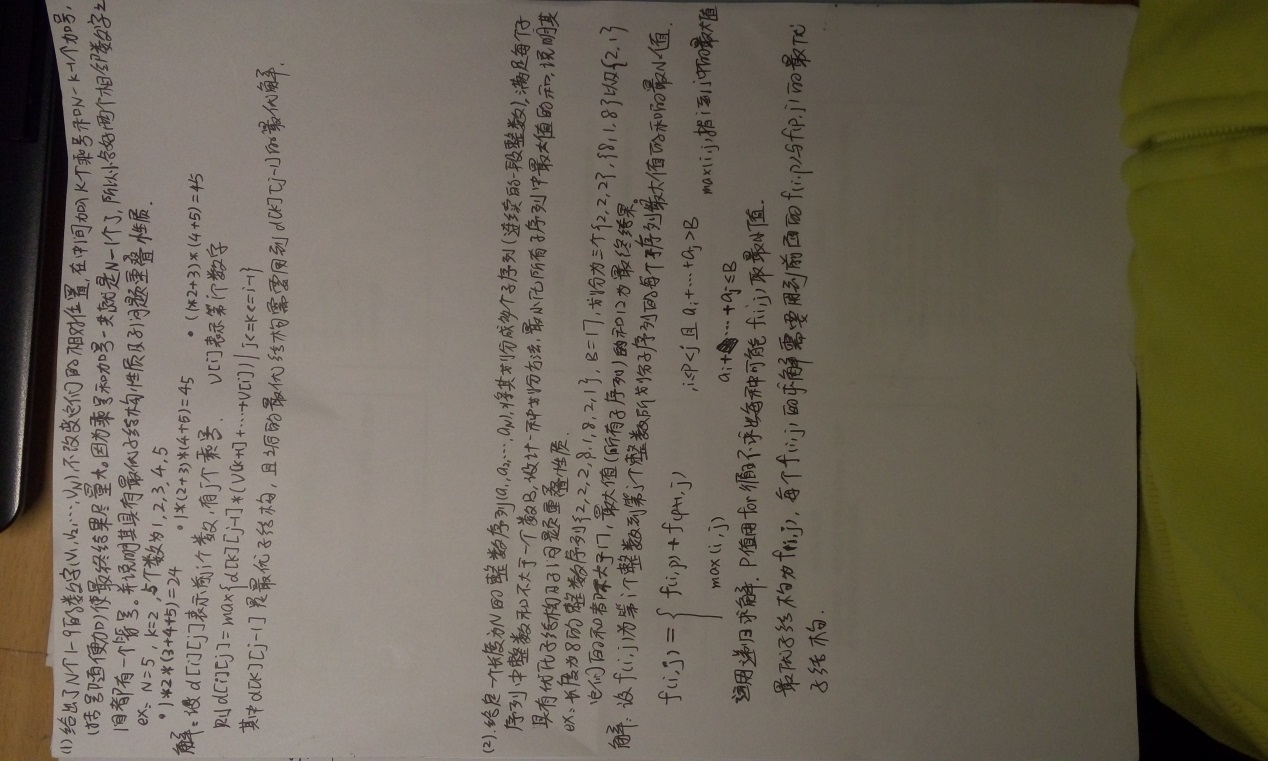
21 cout << F[N][K];

22 return 0;

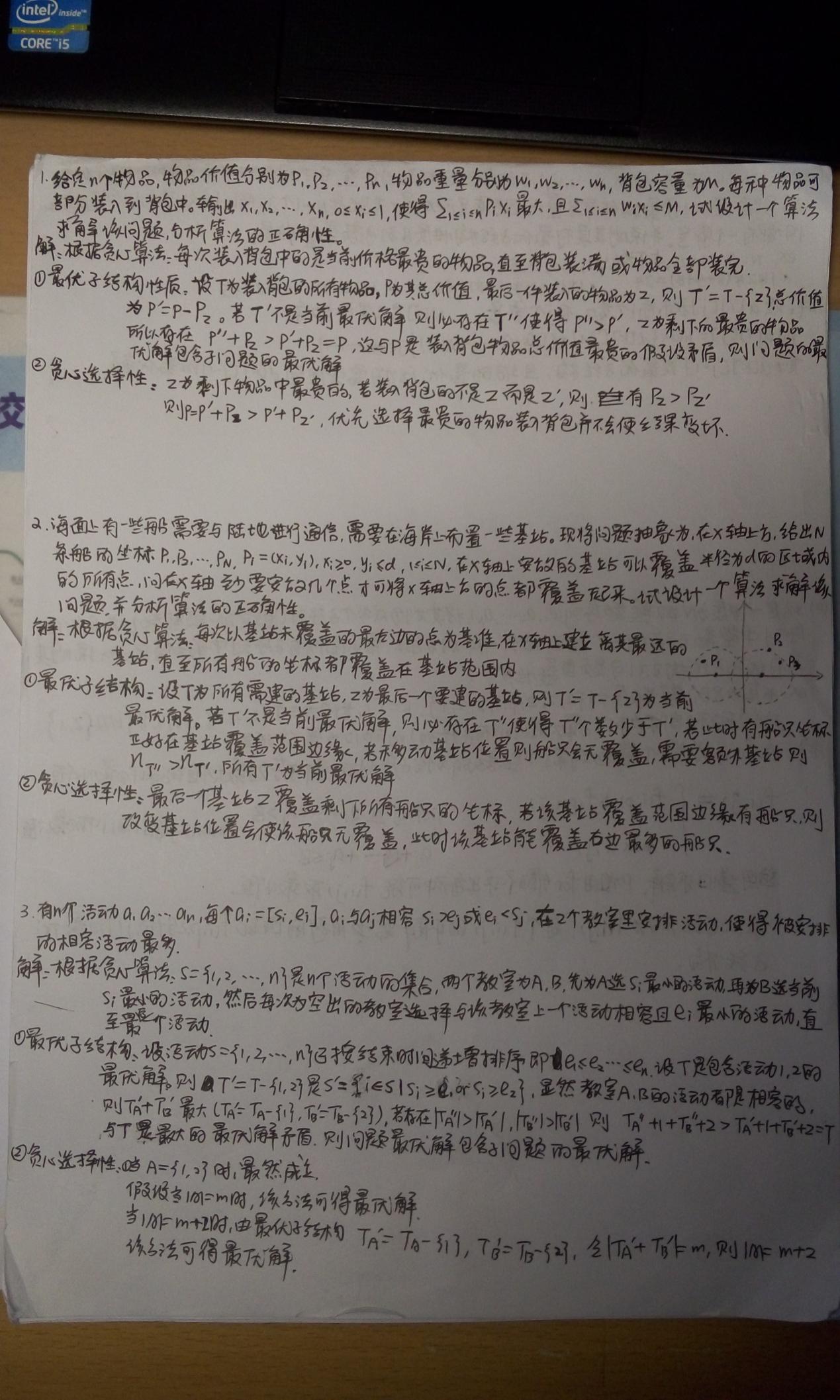
23 }

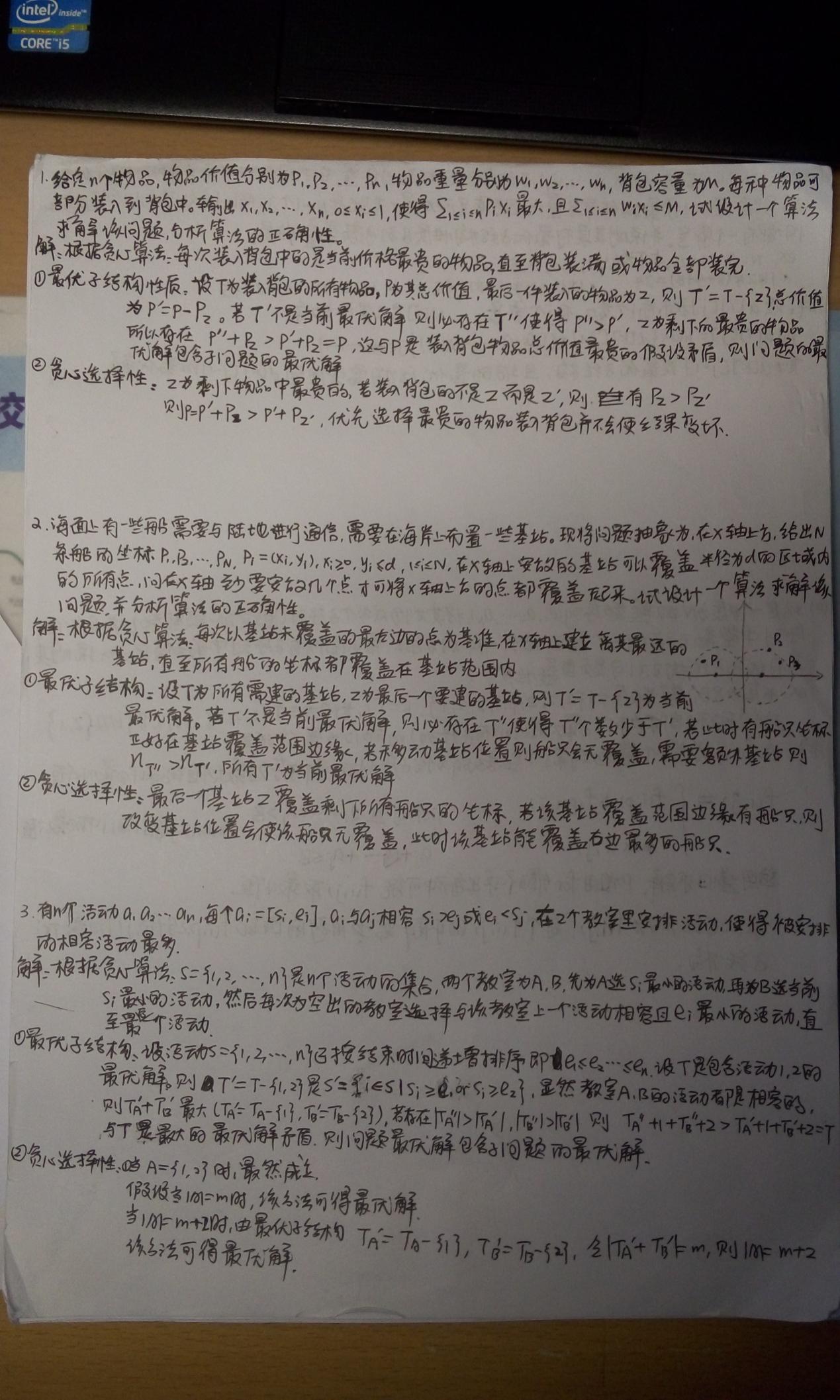
2、给定一长度为N的整数序列（a1、a2、……aN），将其划分成多个子序列（此问题中子序列是连续的一段整数），满足每个子序列中整数的和不大于一个数B，设计一种划分方法，最小化所有子序列中最大值的和。

例如：序列长度为8的整数序列2，2，2，8，1，8，2，1，B=17，可将其划分成3个子序列（2，2，2）、（8，1，8）、（2，1），则可满足每个子序列中整数和不大于17，所有子序列中最大值的和12为最终结果

****

**贪心**

****

****