F.Naptime 解题报告

题意：把一天的时间分成 4000 份，你要从中挑取 m 份用来 xxx。不同的时间 xxx 得到的~~快感~~不同，并且每段连续 xxx 的第一份时间是准备时间，没有收益。

很简单的 dp，dp[i][j][2]代表前 i 个取了 j 个，第 i 个取没取。唯一的特殊情况就是 1 和 n 都取的时候 1 也算，就分 1 取没取两种情况分开 dp。注意加上滚动数组。

代码实现的比较丑陋。排版一直很混乱，干脆改成图片贴上来。

```cpp
#include <iostream>
#include <cstdio>
#include <string.h>
#include <string>
#include <algorithm>
using namespace std;
typedef long long ll;

ll dp1[2][4000][2],dp2[2][4000][2];
int a[4000];
int n,m;

int main(){
    while (scanf("%d%d",&n,&m)==2){
        for (int i=1;i<=n;++i) scanf("%d",&a[i]);
        memset(dp1,-1,sizeof(dp1));
        memset(dp2,-1,sizeof(dp2));

        dp1[1][0][0]=0;
        int now = 0;
        for (int i=1;i<n;++i){
            for (int j=0;j<=m;++j) dp1[now][j][0]=dp1[now][j][1]=-1;
            for (int j=0;j<=m;++j){
                if (dp1[1-now][j][0]>=0){
                    dp1[now][j][0]=max(dp1[now][j][0],dp1[1-now][j][0]);
                    dp1[now][j+1][1]=max(dp1[now][j+1][1],dp1[1-now][j][0]);
                }
                if (dp1[1-now][j][1]>=0){
                    dp1[now][j][0]=max(dp1[now][j][0],dp1[1-now][j][1]);
                    dp1[now][j+1][1]=max(dp1[now][j+1][1],dp1[1-now][j][1]+a[i+1]);
                }
            }
            now=1-now;
        }
        dp2[1][1][1]=0;
        now = 0;
        for (int i=1;i<n;++i){
            for (int j=0;j<=m;++j) dp2[now][j][0]=dp2[now][j][1]=-1;
            for (int j=0;j<=m;++j){
                if (dp2[1-now][j][0]>=0){
                    dp2[now][j][0]=max(dp2[now][j][0],dp2[1-now][j][0]);
                    dp2[now][j+1][1]=max(dp2[now][j+1][1],dp2[1-now][j][0]);
                }
                if (dp2[1-now][j][1]>=0){
                    dp2[now][j][0]=max(dp2[now][j][0],dp2[1-now][j][1]);
                    dp2[now][j+1][1]=max(dp2[now][j+1][1],dp2[1-now][j][1]+a[i+1]);
                }
            }
            now=1-now;
        }
        ll ans = max(dp1[1-now][m][0],dp1[1-now][m][1]);
        ans = max(ans,dp2[1-now][m][1]+a[1]);
        ans = max(ans,dp2[1-now][m][0]);
        cout << ans << endl;
    }

    return 0;
}
```