干草堆

一. 考察内容：

动态规划

二. 题目分析：

[题目大意]

按照顺序给你n个草堆，你需要按照顺序从低到高将他们垒起来，每层的草堆宽度不能比下一层更低，求出垒出的最高层数。

[写题思路]

首先我们很容易想到一种方法，贪心！

具体做法，我们可以倒序枚举每个草堆，记录一下当前层草堆的累积宽度和当前草堆高度，最后将草堆垒完即可，对于剩下的不能垒出新的一层的草堆，我们考虑把他们并入底层。

但是~~很显然~~，这种做法是有一些问题的，我们并没有理由将最后多出来的草堆并处底层后不去理会其他草堆的位置关系，因为当我们将多余的操作放入最后一层时，最后一层的草堆变宽了，我们有理由将部分草堆放入上面一层，以此类推，草堆的位置并不一定是最优的，实际测试，这种贪心可以通过47%的测试数据。

我们想到了更为可靠的方法，动态规划，用严格的状态转移代替不一定靠谱的贪心做法。

设状态f[i][j]为前i个草堆，最后j个草堆排成一行的最大高度，考虑O(n^3)的转移，f[i][j]=max(f[i][k])其中k满足sum(k~j-1)≥sum(j,i)。但是这样还是有些慢，我们考虑结合贪心的思想和DP的方法，从后向前扫一遍，维护f[i]为后i个草堆的最小宽度，ans[i]为后i个草堆的最大高度，但是这种转移是n^2的，还是不能满足要求，我们考虑对于转移进行单调队列优化，即可O(n)实现。

三. 代码实现：

```
#define _CRT_SECURE_NO_DEPRECATE
/************************
*创建时间：2018 09 03
*文件类型：源代码文件
*题目来源：BZOJ
*当前状态：已通过
*备忘录：简单贪心
*作者：HtBest
************************/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
// #include <sys/wait.h>
// #include <sys/types.h>
// #include <unistd.h>
using namespace std;
#define MAXN 100001
int n,a[MAXN],f[MAXN],sum[MAXN],now,ans[MAXN];
struct DDDL
{
    int q[MAXN],head,tail;
```

```
    void push(int x)
    {
        while(head<=tail&&sum[x-1]-f[x]>=sum[q[tail]-1]-f[q[tail]]) tail--;
        q[++tail]=x;
    }
    void update(int x)
    {
        while(head<tail&&sum[q[head+1]-1]-f[q[head+1]]>=sum[x-1]) head++;
    }
}q;
/* Variable explain:

*/
void read()
{
    scanf("%d",&n);
    for(int i=1;i<=n;++i)scanf("%d",&a[i]),sum[i]=a[i]+sum[i-1];
    return;
}
int main()
{
    // freopen(".in","r",stdin);
    // freopen(".out","w",stdout);
    read();
    q.head=q.tail=1;
    q.q[1]=n+1;
    for(int i=n;i>=1;i--)
    {
        //f[i]:后i个草堆的最小宽度
        //ans[i]:后i个草堆的最大高度
        q.update(i);
        int nowmin=q.q[q.head];
        f[i]=sum[nowmin-1]-sum[i-1];
        ans[i]=ans[nowmin]+1;
        q.push(i);
    }
    printf("%d\n",ans[1]);
    return 0;
}
```

<题目跳转> <查看代码>