

A. The Enchanted Forest

- [ad](#)

在一个森林中采蘑菇。

可以选择在任意一点处为起点出发。每一分钟之后，这里的蘑菇生一个。

即

进入某一分钟时间段时，有如下过程，

在原先的点中选择相邻的点。

采完到达点的蘑菇。（当前点的蘑菇数为0）

当前点的蘑菇数加1。

生长思考。

- 分类讨论，尝试找出最优策略。
 - 如果 $k \leq n$
 - 直接走长度为k的最大段。
 - 感受其最优性。 a ：原来的蘑菇。 b ：后面长出的蘑菇。
 - 显然对于b上最优策略是，一直向前走。a上的最优策略是选择区间和最大的那一段。显然上面的方案都满足最优。所以整体上的方案应该是最优的。
 - 如果 $k > n$
 - 最优策略是，等待韭菜长起来。一波最后的时间段内，一次性的收割完。
 - 感受其最优性
 - 无论如何走完了a都是最大的。
 - 对于b的角度而言，探究遗留多少的问题。刚走过的连续段中，都是为1.找一个一一匹配都是更大的。所以最小的遗留情况是，连续走完一整段。

//分成两部分来答。

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const int maxn = 2e5 + 10;

int a[maxn];
ll sum[maxn];

void solve()
{
    ll n, k;
    cin >> n >> k;
    for (int i = 1; i <= n; i++)
    {
        cin >> a[i];
        sum[i] = sum[i - 1] + a[i];
    }
    ll ans = 0;
    if (k <= n)
    {
        for (int i = k; i <= n; i++)
```

```

        ans = max(ans, sum[i] - sum[i - k]);
        ans += (k - 1) * k / 2;
    }
    else //策略是明显的，剩下的是分析策略的计算问题。
    {
        ans = sum[n] + (k - 1 + k - n) * n / 2;
    }
    cout << ans << '\n';
}

int main()
{
    ios::sync_with_stdio(false);
    cin.tie(nullptr), cout.tie(nullptr);
    int t;
    cin >> t;
    while (t--)
        solve();
}

```

生长思考

- 对于过程相关的一个统计问题。一般和等差数列，等比数列相关。
- 明确过程，模拟并且计算一个过程。然后就可以更好的感知到了。