

CS35X: Competitive Programming

Lecture 9: Prefix Sums

Joshua Brody

Logistics:
This is the last class!

Problem debrief: Shortestpath1

Sample Problem: Range Sums

- Input: array **A** of up to 100000 integers
 - **A = [-2 15 -14 60 -5 1]**
- Goal: compute maximal value of a subarray $A[l \dots j]$: **$A[i] + \dots + A[j]$**

Sample Problem: Range Sums

- Input: array **A** of up to 100000 integers
 - **A = [-2 15 -14 60 -5 1]**
- Goal: compute maximal value of a subarray $A[i..j]$: **$A[i] + \dots + A[j]$**
- **$A[1..3] = 15 - 14 + 60 = 61$**

Sample Problem: Range Sums

- Input: array **A** of up to 100000 integers
 - **A = [-2 15 -14 60 -5 1]**
- Goal: compute maximal value of a subarray $A[i..j]$: **$A[i] + \dots + A[j]$**
- **$A[1..3] = 15 - 14 + 60 = 61$**
- Brute Force:
 - for each (i,j) , compute **$A[i] + \dots + A[j]$** , return maximal sum
 - **$O(n^3)$** time

Prefix Sums: range sums in $O(n)$ time

- Given an array A , the prefix sum of A is defined such that
 - $\text{pre}[i]$ is the sum of the first i elements of A .

```
pre[0]=0;
```

```
for i = 1..N:
```

```
    pre[i] = pre[i-1]+A[i-1];
```

Prefix Sums: range sums in $O(n)$ time

- Given an array A , the prefix sum of A is defined such that
 - $\text{pre}[i]$ is the sum of the first i elements of A .

```
pre[0]=0;
```

```
for i = 1..N:
```

```
    pre[i] = pre[i-1]+A[i-1];
```

- We can use prefix sums to compute range sums:

```
lo_pref_sum=0;
```

```
hi_rangesum=0;
```

```
for i =1..N:
```

```
    hi_rangesum = max(hi_rangesum, pre[i]-lo_pref_sum)
```

```
    lo_pref_sum = min(lo_pref_sum, pre[i])
```

Kattis Problem: Commercials