Find Number of Frog Hops

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X | D | D | D | D | D | D | = | X+(ANS)D | = Y |
|  |  |  |  |  |  |  |  |  |  |

Y = X + (ANS) \* D

ANS= (Y – X)/D

Find Missing Element in Permutation:

Efficiency 1: Average

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | N-3 | N-2 | N-1 | N | N+1 | = (N+1)! / X | = A! |  |
|  |  |  |  |  |  |  |  |  |  |

A! = (N+1)!/X

X = (N+1)!/A!

Efficiency 2: Model Solution Above average

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | N-3 | N-2 | N-1 | N | N+1 | = (N+1)! - X | = A! |  |
| N+1 | N | N-1 | N-2 | N-3 | 2 | 1 | Rotated Left |  |  |
| N+2 | N+2 | N+2 | N+2 | N+2 | N+2 | N+2 | Sum of 2 rows above |  |  |

There are N columns

Therefore total sum = ( N\*(N+2))/2

TapeEquilibrium:

int solution(int A[], int N);

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A[0] | A[1] |  |  |  | A[P-1] |  |  |  |  |
| A[P] | A[P+1] | A[P+2] |  |  | A[N-1] |  |  |  |  |
| 1 | 2 |  | P |  | P |  |  |  |  |

Split1 = A[0]+A[1]+….+A[P-1]

Passing Cars

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A[0] | A[1] | A[2] | A[3] | A[4] |  |  |  |  |  |
| 0 | 1 | 0 | 1 | 1 |  |  |  |  |  |
| A0+A1 | 1 | A2+A3 | 1 |  |  |  |  |  |  |
| A0+A2 | 0 | A2+A4 | 1 |  |  |  |  |  |  |
| A0+A3 | 1 |  |  |  |  |  |  |  |  |
| A0+A4 | 1 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Number\_of\_pairs\_A[i] = A[i] + [A[i+1]+A[i+2]+…….+A[N-1]]

For loop to put prefix sums in an array

For loop to calculate number of pairs at each qualifying prefix.

Predict O(n+m) time complexity

Genomic Range Query

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P[0] | P[1] | P[2] | Q[0] | Q[1] | Q[2] |  |  |  |  |
| 2 | 5 | 0 | 4 | 5 | 6 |  |  |  |  |
| C | A | G | C | C | T | A | S[P[K]] |  |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | P.Q[] |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | A | C | G | T |
|  |  |  |  |  |  | 1 | 2 | 3 | 4 |

MinAvgTwoSlices:

Task Approach:

N = size of array

P[N-1] = A[0]+A[1]+….+A[N-1]

P[i] = p[i-1] +A[i]

Slice2[0] = A0+A1

Slice2[1]= A1+A2

Slice2[2] = A2+A3

Slice2[3] = A3+A4

Slice2[4] = A4+A5

Slice2[5] = A5+A6

Slice2[i] = A[i] + A[i+1]

or

Slice2[i] = P[i+1] – p[i-1]

Slice3[0] = A0+A1+A2

Slice3[1] = A1+A2+A3

Slice3[2] = A2+A3+A4

Slice3[3] = A3+A4+A5

Slice3[4] = A4+A5+A6

Slice3[i] = A[i] + A[i+1] + A[i+2]

or

Slice3[i] = P[i+2] – P[i-1]

1. For loop to calculate Prefix Sum A[0] …A[N-1]
2. For loop to calculate 2 index sums
3. For loop to calculate 3 index sums
4. For loop to compare 2& 3 index for lowest value