813. Largest Sum of Averages

DescriptionHintsSubmissionsDiscussSolution

- Difficulty:Medium
- Total Accepted:1.1K
- Total Submissions:3.2K
- Contributor:awice
-

- Subscribe to see which companies asked this question.

Related Topics Dynamic Programming + DFS

We partition a row of numbers A into at most K adjacent (non-empty) groups, then our score is the sum of the average of each group. What is the largest score we can achieve?

Note that our partition must use every number in A, and that scores are not necessarily integers.

```
Example:
Input:
A = [9,1,2,3,9]
K = 3
Output: 20
Explanation:
The best choice is to partition A into [9], [1, 2, 3], [9]. The answer is 9 + (1 +
2 + 3) / 3 + 9 = 20.
We could have also partitioned A into [9, 1], [2], [3, 9], for example.
That partition would lead to a score of 5 + 2 + 6 = 13, which is worse.
```

Note:

- `1 <= A.length <= 100`.
- `1 <= A[i] <= 10000`.
- `1 <= K <= A.length`.
- Answers within `10^-6` of the correct answer will be accepted as correct.

Seen this question in a real interview before?

Let dp[idx][k] be the max average at index idx with remaining k partitions.

```
double maxAvg(vector<int> nums, int idx, int k, vector<vector<double>> &dp)
{
	if(k==0 && idx<nums.size()) return -9999999999;
	if(k==0 && idx==(int)nums.size()) return 0;
	if(dp[idx][k]!=-1.0) return dp[idx][k];
	double avg=-9999999999; double sum = 0;
	for(int i=idx;i<nums.size();++i)
	{
		sum+=nums[i];
		double avg_tmp = sum/(i-idx+1) + maxAvg(nums,i+1,k-1,dp);
		avg = max(avg,avg_tmp);
	}
	dp[idx][k] = avg;
	return avg;
}

double largestSumOfAverages(vector<int>& A, int K) {
	int sz = A.size();
	vector<vector<double>> dp(sz+1,vector<double>(K+1,-1.0));
	return maxAvg(A,0,K,dp);
}
```