

Maximum path sum for each position with jumps under divisibility condition

给定数组，从1号位置开始，每次可以跳到下一个位置（前提是下一个位置X % 当前位置 = 0），输出跳至数组每个位置的方案中和最大的方案

对于索引*i*来说，其代表的位置为 *i* + 1：

从*j* = 1开始to *j* < *i* + 1为止（*j*代表位置，转换为索引为*j*-1）

当(*i* + 1) % *j* == 0 && (*i* + 1) != *j*时：状态转移方程为：

$$dp[i] = \max(dp[i], dp[j - 1] + arr[i]);$$

```
#include <bits/stdc++.h>

using namespace std;

void maxPath(vector<int> &arr, int &N) {
    vector<int> dp(N);
    for (int i = 0; i < N; ++i) {
        dp[i] = arr[i];
        for (int j = 1; j < (i + 1); ++j) {
            if ((i + 1) % j == 0 && (i + 1) != j) {
                dp[i] = max(dp[i], dp[j - 1] + arr[i]);
            }
        }
    }
    for (int i = 0; i < N; ++i) {
        !i ? printf("%d", dp[i]) : printf(" %d", dp[i]);
    }
    printf("\n");
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N;
        scanf("%d", &N);
        vector<int> arr(N);
        for (int i = 0; i < N; ++i) scanf("%d", &arr[i]);
        maxPath(arr, N);
    }
}
```