Cutted Segments

DP

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

using namespace std;

// dp[i] 表示长度i需要切几次，如果长度i不能切分，则为-1
int cutRod(int len, vector<int>& dp, int x, int y, int z) {
    dp[0] = 0;
    int start = min(x, min(y, z));
    for (int i = start; i <= len; ++i) {
        // 如果当前长度i可以切x且切完后剩余的长度仍然存在解决方案，则可以切掉x
        if (i - x >= 0 && dp[i - x] != -1)
            dp[i] = max(dp[i], 1 + dp[i - x]);
        // 如果当前长度i可以切y且切完后剩余的长度仍然存在解决方案，则可以切掉y
        if (i - y >= 0 && dp[i - y] != -1)
            dp[i] = max(dp[i], 1 + dp[i - y]);
        // 如果当前长度i可以切z且切完后剩余的长度仍然存在解决方案，则可以切掉z
        if (i - z >= 0 && dp[i - z] != -1)
            dp[i] = max(dp[i], 1 + dp[i - z]);
    }
    return dp[len];
}
int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int len, x, y, z;
        scanf("%d %d %d %d", &len, &x, &y, &z);
        vector<int> dp(len + 1, -1);
        printf("%d\n", cutRod(len, dp, x, y, z));
    }
}
```

递归 (TLE)

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

using namespace std;

int getCut(int N, int x, int y, int z) {
    if (N <= 0) return 0;
    if (N - x < 0 && N - y < 0 && N - z < 0) return 0;
    int count = 0;
    count = 1 + max(getCut(N - x, x, y, z), max(getCut(N - y, x, y, z), getCut(N - z, x, y, z)));
    return count;
}

int cutRod(int N, int x, int y, int z) {
    if (N <= 0) return 0;
    if (N - x < 0 && N - y < 0 && N - z < 0) return 0;
//    printf("%d %d %d\n", 1 + getCut(N - x, x, y, z), 1 + getCut(N - y, x, y, z), 1 + getCut(N - z, x, y, z));
    return max(1 + getCut(N - x, x, y, z), max(1 + getCut(N - y, x, y, z), 1 + getCut(N - z, x, y, z)));
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N, x, y, z;
        scanf("%d %d %d %d", &N, &x, &y, &z);
        printf("%d\n", cutRod(N, x, y, z));
    }
}
```