

Zıp zıp kurbağa bir nehri karşıdan karşıya geçmek istemektedir. Nehrin iki kenarı arasında N adet taş vardır. Kurbağa sadece ileri doğru ve tek bir seferde rastgele olarak 1, 2 veya 3 sonraki taşla zıplamaktadır. N adet taş verildiğinde nehrin bir kenarından diğer kenarına kaç farklı şekilde ulaşabileceği recursive ve döngü ile bulan algoritmalar.



Örneğin 1 taş varsa sadece 1 farklı yol vardır. 2 taş varsa 1+1 ve 2 olarak 2 yol vardır. 3 taş varsa 1+1+1, 1+2, 2+1 ve 3 olarak 4 farklı yol vardır. 4 taş varsa 1+1+1+1, 1+1+2, 1+2+1, 1+3, 2+1+1, 2+2 ve 3+1 şeklinde 7 yol vardır.

Döngü ile hesaplayan algoritma:

```
long long iterativeFunction(int numberOfStones)
{
    vector<long long> arr(numberOfStones);
    arr[0]=1; //Possibilities when there is 1 stone:1
    arr[1]=2; //Possibilities when there is 2 stone:1+1,2
    arr[2]=4; //Possibilities when there is 3 stone:1+1+1,1+2,2+1,3
    for(int i=3;i<numberOfStones;i++)
        {arr[i]=arr[i-1]+arr[i-2]+arr[i-3];}
    /*
        i tane taş için ihtimal sayısı=
        i-1 tane taş için ihtimal sayısı+
        i-2 tane taş için ihtimal sayısı+
        i-3 tane taş için ihtimal sayısı
    */
    return arr[numberOfStones-1];
}
```

Ekran çıktısı:

```
iterative 1 -> 1 ( 3us )
iterative 2 -> 2 ( 1us )
iterative 3 -> 4 ( 1us )
iterative 4 -> 7 ( 0us )
iterative 5 -> 13 ( 1us )
iterative 6 -> 24 ( 0us )
iterative 7 -> 44 ( 6us )
iterative 8 -> 81 ( 1us )
iterative 9 -> 149 ( 1us )
iterative 10 -> 274 ( 1us )
iterative 11 -> 504 ( 1us )
iterative 12 -> 927 ( 1us )
iterative 13 -> 1705 ( 1us )
iterative 14 -> 3136 ( 1us )
iterative 15 -> 5768 ( 2us )
iterative 16 -> 10609 ( 1us )
iterative 17 -> 19513 ( 2us )
iterative 18 -> 35890 ( 1us )
iterative 19 -> 66012 ( 2us )
iterative 20 -> 121415 ( 2us )
iterative 21 -> 223317 ( 2us )
iterative 22 -> 410744 ( 2us )
iterative 23 -> 755476 ( 2us )
iterative 24 -> 1389537 ( 2us )
iterative 25 -> 2555757 ( 2us )
iterative 26 -> 4700770 ( 2us )
iterative 27 -> 8646064 ( 2us )
iterative 28 -> 15902591 ( 2us )
iterative 29 -> 29249425 ( 2us )
iterative 30 -> 53798080 ( 2us )
iterative 31 -> 98950096 ( 8us )
iterative 32 -> 181997601 ( 3us )
iterative 33 -> 334745777 ( 3us )
iterative 34 -> 615693474 ( 2us )
iterative 35 -> 1132436852 ( 3us )
iterative 36 -> 2082876103 ( 3us )
iterative 37 -> 3831006429 ( 3us )
iterative 38 -> 7046319384 ( 3us )
iterative 39 -> 12960201916 ( 3us )
iterative 40 -> 23837527729 ( 3us )
iterative 41 -> 43844049029 ( 3us )
iterative 42 -> 80641778674 ( 3us )
iterative 43 -> 148323355432 ( 3us )
iterative 44 -> 272809183135 ( 3us )
iterative 45 -> 501774317241 ( 3us )
iterative 46 -> 922906855808 ( 3us )
iterative 47 -> 1697490356184 ( 3us )
iterative 48 -> 3122171529233 ( 3us )
iterative 49 -> 5742568741225 ( 4us )
iterative 50 -> 10562230626642 ( 4us )
iterative 51 -> 19426970897100 ( 4us )
iterative 52 -> 35731770264967 ( 4us )
iterative 53 -> 65720971788709 ( 4us )
iterative 54 -> 120879712950776 ( 4us )
iterative 55 -> 222332455004452 ( 4us )
iterative 56 -> 408933139743937 ( 4us )
iterative 57 -> 752145307699165 ( 4us )
iterative 58 -> 1383410902447554 ( 4us )
iterative 59 -> 2544489349890656 ( 5us )
iterative 60 -> 4680045560037375 ( 4us )
iterative 61 -> 8607945812375585 ( 5us )
iterative 62 -> 15832480722303616 ( 5us )
iterative 63 -> 29120472094716576 ( 5us )
iterative 64 -> 53560898629395777 ( 5us )
iterative 65 -> 98513851446415969 ( 5us )
iterative 66 -> 181195222170528322 ( 5us )
iterative 67 -> 333269972246340068 ( 5us )
iterative 68 -> 612979045863284359 ( 5us )
iterative 69 -> 1127444240280152749 ( 5us )
iterative 70 -> 2073693258389777176 ( 5us )
iterative 71 -> 3814116544533214284 ( 5us )
iterative 72 -> 7015254043203144209 ( 5us )
```

Ekran çıktısının böyle olmasının sebebi:Fonksiyona gönderilen taş sayısını için kaç ihtimal olduğunun hesaplanması (1,2,3 taş hariç) kendinden önceki 3 taşla kaç farklı şekilde gelinebildiğinin toplamıdır. Yukarıdaki algoritmada taş sayıları için kaç ihtimal olduğunu vector'e kaydettiğinden bellekten bir miktar kayıp ile hızdan büyük bir kazanç elde ediliyor.

Recursive şekilde hesaplayan algoritma:

```
long long recursiveFunction(int numberOfStones)
{
    switch (numberOfStones)
    {
        case 1: return 1;//1
        case 2: return 2;//1+1,2
        case 3: return 4;//1+1+1,1+2,2+1,3
        default:
            return recursiveFunction(numberOfStones-1)+
                recursiveFunction(numberOfStones-2)+
                recursiveFunction(numberOfStones-3);

        /*
            i tane taş için ihtimal sayısı=
            i-1 tane taş için ihtimal sayısı+
            i-2 tane taş için ihtimal sayısı+
            i-3 tane taş için ihtimal sayısı
        */
    }
}
```

Ekran çıktısı:

recursive 1 -> 1 (0us)	recursive 21 -> 223317 (724us)
recursive 2 -> 2 (0us)	recursive 22 -> 410744 (1185us)
recursive 3 -> 4 (0us)	recursive 23 -> 755476 (2223us)
recursive 4 -> 7 (0us)	recursive 24 -> 1389537 (5469us)
recursive 5 -> 13 (0us)	recursive 25 -> 2555757 (6111us)
recursive 6 -> 24 (0us)	recursive 26 -> 4700770 (16867us)
recursive 7 -> 44 (0us)	recursive 27 -> 8646064 (37092us)
recursive 8 -> 81 (0us)	recursive 28 -> 15902591 (59585us)
recursive 9 -> 149 (0us)	recursive 29 -> 29249425 (108288us)
recursive 10 -> 274 (1us)	recursive 30 -> 53798080 (185252us)
recursive 11 -> 504 (1us)	recursive 31 -> 98950096 (401464us)
recursive 12 -> 927 (3us)	recursive 32 -> 181997601 (654490us)
recursive 13 -> 1705 (6us)	recursive 33 -> 334745777 (1620840us)
recursive 14 -> 3136 (12us)	recursive 34 -> 615693474 (2344053us)
recursive 15 -> 5768 (22us)	recursive 35 -> 1132436852 (4303954us)
recursive 16 -> 10609 (35us)	recursive 36 -> 2082876103 (7497631us)
recursive 17 -> 19513 (69us)	recursive 37 -> 3831006429 (12893336us)
recursive 18 -> 35890 (113us)	recursive 38 -> 7046319384 (28940696us)
recursive 19 -> 66012 (210us)	recursive 39 -> 12960201916 (53027261us)
recursive 20 -> 121415 (423us)	recursive 40 -> 23837527729 (94941630us)

Ekran çıktısının böyle olmasının sebebi:Fonksiyona gönderilen taş sayısını için kaç ihtimal olduğunun hesaplanması (1,2,3 taş hariç) kendinden önceki 3 taşa kaç farklı şekilde gelinebildiğinin toplamıdır. Yukarıdaki algoritmada taş sayıları için kaç ihtimal olduğunu recursive şekilde hesapladığı ve her

değer için 3 recursive fonk çağrıldığı için taş sayısı arttıkça hesaplama süresi büyük ölçüde artmaktadır.Bu nedenle 40 civarında hesaplama takılıyor ve sonuç bulması çok uzun sürüyor.