1

(1)不妨设胜利的概率$p=x$

$P(i,j)=xP(i-1,j)+(1-x)P(i,j-1)$

$\forall j,i\in[0,size]\quad P(0,j)=1\quad P(i,0)=0$

(2)$According\quad to\quad seven-game$(即七局四胜)

即要求$P(4,4)$的值，$P_{Awin}=P(4,4)=0.289792=28.98\%$

```
double solution(const int n, const double p)//n为场次，p为A获胜的概率
{
    int size = (n + 1) / 2;
    vector<vector<double>> arr(size + 1, vector<double>(size + 1, 0));
    for (int i = 0; i < size + 1; i++)
    {
        arr[0][i] = 1;
        arr[i][0] = 0;
    }
    for (int i = 1; i < size + 1; i++)
    {
        for (int j = 1; j < size + 1; j++)
        {
            arr[i][j] = p * arr[i - 1][j] + (1 - p) * arr[i][j - 1];
        }
    }
    return arr[size][size];
}
int main()
{
    int n = 7;
    double p = 0.4;
    cout << solution(n, p);
}
```

(3)

$pseudocode$(伪代码表示)

时间复杂度：初始化时间复杂度是$O(size) = O(\frac{n}{2})$,赋值的时间复杂度是$O((size-1)*(size-1)*2) = O(\frac{n^2}{2})$,所以总的时间复杂度为$O(\frac{n^2}{2}+\frac{n}{2}) = O(n^2)$

空间复杂度:由于只创建了一个大小为$(size+1)*(size+1)$的二维数组,所以空间复杂度为$O(n^2)$

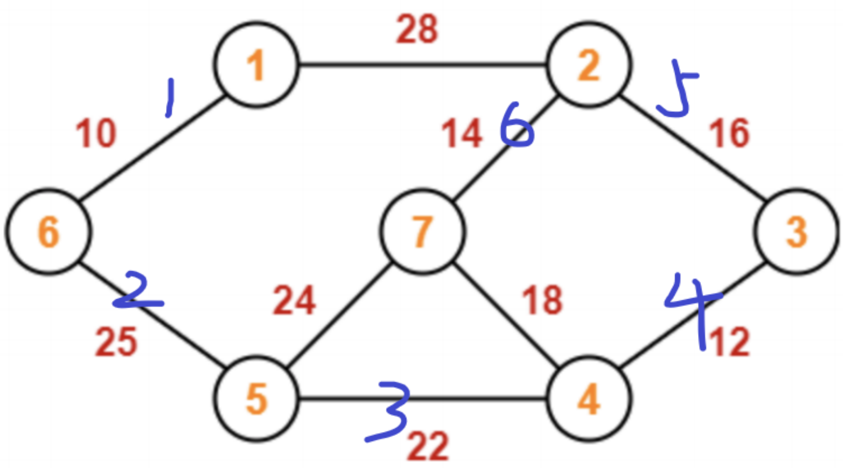
```
double:solution(integer:n,double:p)//n为场次，p为A获胜的概率
{
    size<-(n+1)/2
    vector<vector<double>>arr(size+1,vector<double>
(size+1,0))
    for i=0 to size+1 do
        arr(0,i)=1
        arr(i,0)=0
    endfor

    for i=1 to size+1
        for j=1 to size+1
            do
            arr(i,j)=p*arr(i-1,j)+(1-p)*arr(i,j-1)
        endfor
    endfor
    Return arr(size,size)
}
int:main()
{
    int:n//场次
    double:p//A获胜概率
    Input:n,p
    Ouput:solution(n,p)
}
```

2

$(1) cost = 10 + 25 + 22 + 12 + 16 + 14 = 99$

a. Using Prim's Algorithm



$(2) cost = 1 + 2 + 3 + 4 + 6 + 10 = 26$

b. Using Kruskal's algorithm