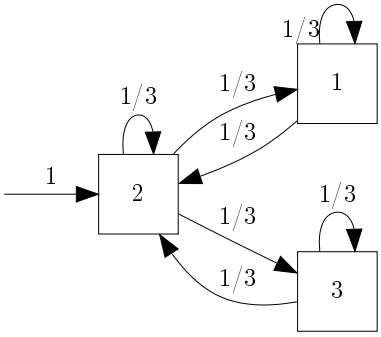
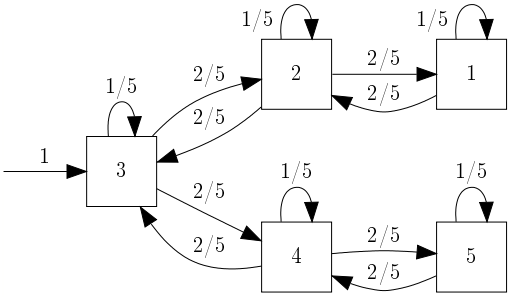
Métodos Numéricos
Trabalho II
Prof. João B. Oliveira

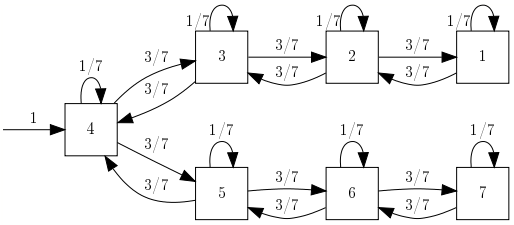
Seu trabalho nesta disciplina consiste em realizar um estudo secreto para um estúdio coreano que está planejando um novo seriado muito viciante e inovador: pessoas aleatórias que precisam de dinheiro são convidadas a participar de um jogo onde coisas ruins podem acontecer com elas. Elas devem ir para um prédio misterioso com n salas conectadas uma com a outra, como nos exemplos abaixo (de três prédios diferentes sendo pensados como cenário):

O simulador de jogo só depende do número n de salas e as regras são estas:

- O número n de salas do cenário é sempre ímpar;
- Os personagens são sempre largados na sala $\lfloor (n+1)/2 \rfloor$;
- A cada episódio os personagens tem probabilidade $1/n$ de continuar na mesma sala (e portanto tem probabilidade $(n-1)/n$ de se mover para as salas dos lados.)
- Se o personagem estiver em uma das salas de 2 até $n-1$, ele tem probabilidade $(n-1)/2$ de ir para a sala que vem antes e a mesma probabilidade de ir para a que vem depois. Nas salas de 2 a n acontecem coisas interessantes, mas os personagens não chegam a sair do jogo.
- As salas das pontas (casas 1 e n) não tem vizinhas em um dos lados, mas os personagens tem probabilidade $(n-1)/2$ de morrer tragicamente e sair do seriado.







O estúdio entende que os personagens largados irão circulando pelas salas ao acaso, indo para um lado e pro outro e gerando um seriado potencialmente infinito, com audiência garantida, uma espécie de BBB com mortes nas pontas. O que eles não tem certeza é de quantos atores serão necessários para ficar andando pra lá e pra cá, pois isso vai mudar com o número de salas no cenário.

Sua missão é construir um programa que seja capaz de receber n pela linha de comando e possa responder a pergunta: quantas pessoas (no total) devem estar nas salas quando o jogo for estabilizado? Então, espera-se que seu simulador rode mais ou menos assim:

```
> java simuleitor 23       // 23 salas
População para 23 salas: 1234 atores    // 1234 parece meio aleatório...
```

Ah, sim. Você **não pode** simular o movimento de pessoas entre as salas. A única alternativa que é aceita para resolver o problema é através de um sistema linear! E para garantir que seu programa pode ser rodado em qualquer estúdio coreano ele deve ser feito em Java, C, Python ou C++. Os estúdios coreanos gostam bastante de C e C++.