

# Universidade Federal de Minas Gerais

Bacharel em Sistemas de Informação  
Algoritmos e Estruturas de Dados 3



Trabalho Prático 1  
Agosto 2017

Gabriel Silva Bastos  
Matrícula: 2016058204

# 1 Introdução

Nubby está de volta, desta vez para organizar um campeonato de *Mortal Kontest*. Participarão  $\{v_1, \dots, v_n\}$  amigos ( $1 \leq n \leq 25$ ), e Nubby ficará encarregado apenas da organização de todas as  $n - 1$  rodadas. Em cada rodada, dois amigos dos que não foram eliminados serão escolhidos aleatoriamente para o embate, e o que perder é eliminado do campeonato. Desta forma, a última rodada será entre os dois amigos restantes, e o vencedor desta é o grande campeão.

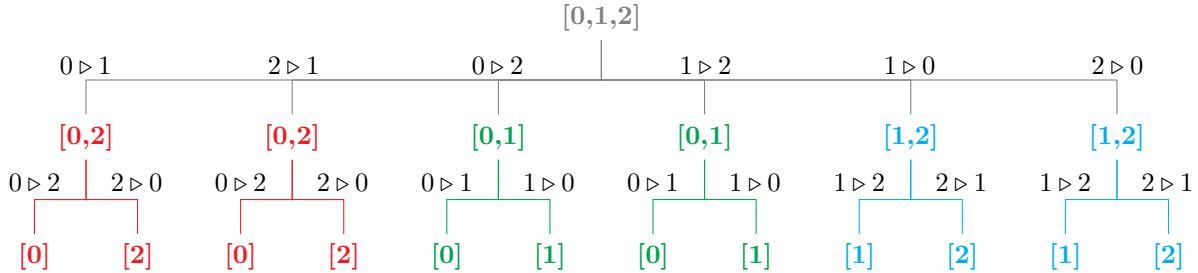
Nubby possui dados de batalhas anteriores entre seus amigos, e através destes calculou a probabilidade de cada amigo vencer os demais em um embate. Através destas probabilidades, Nubby quer calcular a probabilidade de cada um de seus amigos vencer o campeonato.

## 2 Visão Geral da Solução

A entrada consiste de um número  $n$  de jogadores, e uma matriz  $V \in \mathbb{M}_{n \times n}$  contendo as probabilidades calculadas por Nubby. A posição  $i, j$  na matriz contém a probabilidade  $V_{i,j} \in [0, 1]$  do jogador  $v_i$  vencer o jogador  $v_j$  em um embate. Além disso, todas posições da forma  $i, i$  contém o valor 0, que é a probabilidade de um jogador vencer à si mesmo, e as demais posições  $i, j$  satisfazem  $V_{i,j} = 1 - V_{j,i}$ .

$$\begin{bmatrix} 0 & \dots & V_{1,n} \\ \vdots & \ddots & \vdots \\ V_{n,1} & \dots & 0 \end{bmatrix}$$

Para ilustrar a solução, utilizaremos um caso particular onde  $n = 3$ . A seguinte árvore representa os possíveis desdobramentos do campeonato neste caso:



Cada nó da árvore corresponde ao grupo  $G$  ainda presente no campeonato.

As folhas contém grupos unitários, que indicam o campeão do desdobramento descrito pelo caminho da raiz até a folha.

O rótulo  $i > j$  para cada aresta indica que na rodada e configuração correspondentes os jogadores  $i$  e  $j$  foram escolhidos para o embate, e o jogador  $i$  venceu.

Cada nível de arestas corresponde à uma rodada, e cada aresta é uma possibilidade para a rodada. O número  $a$  de escolhas de dois jogares é

$$a = \binom{|G|}{2} = \frac{|G|^2 - |G|}{2} \quad (1)$$

Portanto, o número de arestas em cada nível é  $2a$ , pois para cada escolha de dois jogadores há dois possíveis vencedores.

Cada caminho da raiz até um nó na árvore descreve um desdobramento do campeonato, e portanto possui uma probabilidade associada para cada possível vencedor. Tais probabilidades são compostas pela combinação de dois itens:

- A probabilidade associada à rodada: Cada rodada é modelada como uma aresta para cada escolha de um vencedor e um perdedor possível. Cada aresta corresponde à equiprovável da escolha dos jogadores vezes a chance do vencedor vencer.

$$P(i \triangleright j) = \frac{1}{a} \cdot V_{i,j} = \frac{2}{|G|^2 - |G|} \cdot V_{i,j}$$

- A probabilidade associada à subárvore conectada pela aresta, que corresponde à combinação das probabilidades das subarestas.

O segundo item denota um subproblema, pois a subárvore conectada por cada aresta pode ser considerada um subcampeonato. Como é notável no diagrama, vários subcampeonatos equivalentes ocorrem quando geramos a árvore de desdobramentos. Estes foram dispostos com a mesma cor para melhor visualização.

Efetivamente, a probabilidade de um jogador vencer o campeonato é o somatório das probabilidades dos desdobramentos em que ele vence.

## 2.1 Estratégia de Memorização

Os subproblemas foram definidos como os subcampeonatos. Um subcampeonato é constituído pela exclusão de alguns membros do campeonato original  $C$ , devido à eliminação das rodadas passadas. Portanto, todos subcampeonatos possíveis pertencem ao conjunto potência de  $C$ .

Como todos os desdobramentos do campeonato  $C$  são necessários para a solução do problema, gerar o conjunto potência se torna necessário. Porém, os subcampeonatos com menos de 2 jogadores não são interessantes, então podemos excluí-los da memorização. Portanto, é necessário memorizar  $m$  grupos

$$\begin{aligned} m &= |\mathcal{P}(C)| - |C| - 1 \\ &= 2^n - n - 1 \end{aligned}$$

Para memorizar tais grupos, um vetor foi adotado. Também foi necessária uma forma eficiente de identificar cada grupo no vetor. Considerando que o tamanho máximo de jogadores definido por Nubby foi 25, um inteiro de largura fixa de 32 bits é utilizado para identificar cada grupo. Cada bit no corresponde à presença do jogador no grupo

$$\begin{aligned} 0 \dots 00011010_2 &\equiv [1, 3, 4] \\ 0 \dots 100100101_2 &\equiv [0, 2, 5, 8] \\ 0 \dots 01111111_2 &\equiv [0, 1, 2, 3, 4, 5, 6, 7] \end{aligned}$$

Para converter desta representação para um índice no vetor, é necessário descontar os grupos unitários e o grupo vazio que não foram considerados. A seguinte função realiza esta compensação em um identificador de grupo  $g$ :

$$\varphi(g) = g - \lceil \log_2 g \rceil - 1 \quad (2)$$

Ao realizar *benchmarks* com programa, o tempo de execução estava bem acima do esperado para entradas grandes. Ao investigar, foi concluído que um grande número de *cache misses*<sup>1</sup> ocorria devido à estratégia<sup>2</sup> de correção dos índices no vetor de memorização. Tal estratégia portanto foi descartada, e aceitou-se o desperdício de memória dos conjuntos unitários e vazio em prol do tempo de execução. Reduções de mais de uma hora na execução foram observadas para entradas de tamanho  $n = 25$ .

$$m = |\mathcal{P}(C)| = 2^n \quad (3)$$

<sup>1</sup>[https://en.wikipedia.org/wiki/CPU\\_cache#Cache\\_miss](https://en.wikipedia.org/wiki/CPU_cache#Cache_miss)

<sup>2</sup>Equação 2

## 3 Análise de Complexidade

### 3.1 Espacial

Para a matriz  $M$ , são alocados  $n^2$  floats.

Para cada subcampeonato, são alocados  $n$  floats, correspondendo à probabilidade de cada jogador vencer. Aos jogadores não presentes no subcampeonato é atribuída probabilidade 0 de vencer.

São  $2^n$  grupos, portanto  $2^n \cdot n$  floats alocados.<sup>3</sup>

A complexidade espacial final é

$$\begin{aligned}\Theta(n^2 + 2^n \cdot n) \\ \Theta(n \cdot 2^n)\end{aligned}$$

### 3.2 Temporal

Todos os subcampeonatos são calculados exatamente uma vez. Portanto, as probabilidades para todos grupos do conjunto potência do campeonato, exceto os unitários e o vazio, serão calculadas.

No conjunto potência, há exatamente  $\binom{n}{i}$  subconjuntos de cardinalidade  $i$ .<sup>4</sup> Cada um destes subconjuntos representa um subgrupo de tamanho  $i$ , nos quais a rodada imediata contém  $2^i$  combinações.<sup>5</sup> Como todos subcampeonatos de cardinalidade 2 até  $n$  serão calculados, a complexidade se resume ao somatório:

$$f(n) = 2 \cdot \sum_{i=2}^n \binom{n}{i} \binom{i}{2}$$

Para desenvolver o somatório, utilizaremos alguns artifícios <sup>6</sup>

$$\begin{aligned}(1+x)^n &= \sum_{i=0}^n \binom{n}{i} \cdot x^i && \text{pelo teorema binomial} \\ n(n-1)(1+x)^{n-2} &= \sum_{i=2}^n \binom{n}{i} \cdot i(i-1) \cdot x^i && \text{derivando ambos lados duas vezes} \\ n(n-1) \cdot 2^{n-2} &= \sum_{i=2}^n \binom{n}{i} \cdot i(i-1) && \text{substituindo } x \text{ por } 1 \\ n(n-1) \cdot 2^{n-2} &= 2 \cdot \sum_{i=2}^n \binom{n}{i} \cdot \frac{i(i-1)}{2} && \text{multiplicando o lado direito por } \frac{2}{2}, \text{ reorganizando} \\ n(n-1) \cdot 2^{n-2} &= 2 \cdot \sum_{i=2}^n \binom{n}{i} \binom{i}{2} && \text{pela definição de } \binom{i}{2}\end{aligned}$$

Portanto, a complexidade obtida é

$$\begin{aligned}\Theta(n(n-1) \cdot 2^{n-2}) \\ \Theta(n^2 - n) \cdot \Theta(2^{n-2}) \\ \Theta(n^2) \cdot \Theta(2^n) \\ \Theta(n^2 \cdot 2^n)\end{aligned}$$

---

<sup>3</sup>Equação 3

<sup>4</sup>[https://en.wikipedia.org/wiki/Power\\_set#Relation\\_to\\_binomial\\_theorem](https://en.wikipedia.org/wiki/Power_set#Relation_to_binomial_theorem)

<sup>5</sup>Equação 1

<sup>6</sup><https://math.stackexchange.com/questions/2472972/prove-that-sum-i-2n-binomni-binomi2-2n-3-cdot-nn-1>

## 4 Análise Experimental

A análise experimental da implementação é mostrada pela figura 1. Para realizar os experimentos, foi feito um gerador de entradas. Para um valor  $n$ , é gerada uma entrada contendo uma matriz  $V \in \mathbb{M}_{n \times n}$ . Para medir o tempo de execução do código, foi utilizada a informação de uso de recursos fornecida pelo sistema operacional linux.<sup>7</sup> A razão inicial é verificar, de forma geral, o comportamento do algoritmo para casos genéricos.

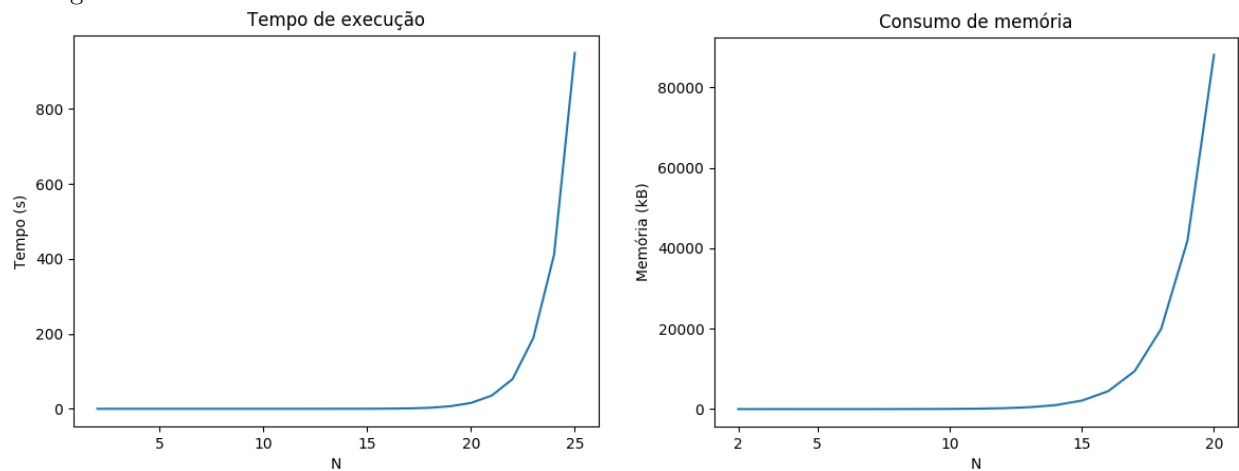


Figura 1: Teste experimental

Nos gráficos da figura comprova-se as complexidades calculadas anteriormente.

## 5 Conclusão

Mesmo a complexidade final sendo superior à exponencial, houve uma grande melhoria em relação à solução inicial desenvolvida sem memorização. O problema em específico se demonstrou interessante no aspecto de que, sem a programação dinâmica, o tempo gasto para a solução é claramente inviável. Portanto, a programação dinâmica não só melhorou o tempo da solução neste caso, como também tornou possível obter resultados antes inviáveis.

---

<sup>7</sup><https://en.wikipedia.org/wiki/Procfs>