

Trabalho Prático 3

Duendes Brigões

Davi Santos Rodrigues - 2022043752

Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brasil

davisrodrigues@ufmg.br

1. Introdução

O presente trabalho abordou a solução de um problema pedagógico sobre o papai noel ter problemas com a sua frota de duendes. O objetivo do trabalho é consolidar os conceitos vistos em sala, sobre programação dinâmica. O problema parece simples, mas esconde uma complexidade assustadora e, também, foi objetivo do trabalho lidar com o problema real.

2. Modelagem

A modelagem desse problema foi muito complicada e de longe o maior desafio. Desde o começo, tentei abordar o problema “Bottom-Up”, mas nenhuma solução parecia ótima, ou correta, ou era mesmo programação dinâmica. Ao ver a dica no enunciado do trabalho, me toquei que ele era mais complexo do que parecia e finalmente percebi que era exponencial. O problema dos duendes pode ser traduzido para o problema do “Maximum Independent Set”, um problema NP-Hard.

Nas minhas tentativas anteriores, eu já havia modelado os duendes como vértices de um grafo com uma aresta não-ponderada entre duendes rivais entre-si e mantive isso na implementação final. Para cada duende há uma lista de adjacência que representa seus vizinhos, os seus inimigos.

Para representar os conjuntos, descobri uma solução elegante e eficiente chamada de Bitmasking. Bitmasking consiste em adereçar um inteiro para representar quais elementos de um conjunto estão presentes em um sub-conjunto. Nas referências coloco o material que foi utilizado para aprender Bitmasking, mas, simplificadamente, utiliza-se bit-operações e shifting para manipular, inserir e unir conjuntos com Bitmasking. Ao final, a resposta é um inteiro cujo a representação em binário indica qual o conjunto de duendes foram selecionados e qual o seu tamanho.

3. Solução

A solução utiliza uma estratégia de programação dinâmica bottom-up chamada de “Sum of Subsets” e foi desenvolvido em 2 partes principais:

Parte 1:

O conjunto de duendes é dividido em 2, lado A e lado B. Conforme a dica, 2^{20} é computável no tempo do testador do problema, o que não é o caso de 2^{40} .

Primeiro o programa processa o lado B, computando as $2^{(n/2)}$ máscaras de conjuntos. Armazena-se essas máscaras em uma tabela de consulta, como dp bottom-up. Utilizando Sum of Subsets, o programa preenche o vetor de máscaras com os conjuntos válidos (os que não possuem duendes brigões). Se a

máscara é válida, seu valor inicial é o número de bits ligados e se não for, ela herda o melhor resultado de seus subconjuntos.

Parte 2:

Para cada uma das $2^{n/2}$ máscaras do lado A, o algoritmo verifica se é uma máscara válida. Depois, acha os vizinhos disponíveis em B invertendo os bits dos vizinhos da máscara atual e confere na tabela gerada pela parte 1 para obter a melhor combinação dos conjuntos.

A solução final é a que maximiza a quantidade de duendes e o critério de desempate é a ordem lexicográfica, que é obtida a comparar o bit menos significativo da diferença das soluções candidatas.

4. Análise de Complexidade

Tempo:

O problema do Maximum Independent Subset é $O(2^n)$. O meu algoritmo ainda tem que passar pela solução contando os bits 1 para determinar o tamanho, portanto é $O(n * 2^n)$.

Espaço:

Para o lado B, criamos uma tabela de máscaras dos conjuntos e também guardamos seu tamanho. Existem $2^{n/2}$, portanto a complexidade de espaço que domina é $O(2^{n/2})$.

5. Considerações Finais

A realização deste trabalho consolidou e aprofundou meus conhecimentos em programação dinâmica e técnicas de otimização. O maior desafio foi implementar o problema e descobrir as ferramentas necessárias para não “explodir” a capacidade de memória e tempo. Tive sorte com as recomendações dos materiais no youtube, onde achei as aulas que foram pilares para a solução desenvolvida. Creio que, da próxima vez que encontrar um problema computacional, não serei tão leviano e analisarei mais profundamente, para perder menos tempo.

6. Referências

Vivek Gupta. “SOS DP Fundamentals and why it works | PART 1 | Learning CP | Concept and Mental Models | English”. 2023. Disponível em: <https://www.youtube.com/watch?v=RYdueyYBQS8>

Kartik Arora. “DP with Bitmasking #1”. Playlist. 2020. Disponível em: https://www.youtube.com/watch?v=685x-rzOIIY&list=PLb3g_Z8nEv1icFNrtZqByO1CrWVHLIO5g&index=3

usaxena95. SOS Dynamic Programming [Tutorial]. Code Forces. 2016. Disponível em: <https://codeforces.com/blog/entry/45223>