Exercícios de Fila de Prioridade Comentados

Daniel Saad Nogueira Nunes

22 de novembro de 2023

Aviso

Os comentários a respeito das soluções das listas de exercício buscam explicar resumidamente a ideia para resolução dos problemas, mas sem "entregar o ouro". O objetivo é apenas fornecer uma direção para aqueles alunos que estejam com dificuldades. Para instruções mais detalhadas, sugiro procurar o professor por e-mail, sala de aula virtual da disciplina ou presencialmente nos horários de atendimento.

Maratonando Cursos

Temos n cursos, cada um descrito por um par (g_i, v_i), os quais representam as semanas de gratuidade do curso e o conhecimento proporcionado pelo curso. O objetivo é alocar os cursos nas m semanas, sem sobreposição, de modo a maximizar a soma dos conhecimentos dos cursos alocados.

Tome uma fila de prioridade P cujo critério seja dar maior prioridade para os cursos de maior conhecimento. A ideia para resolver o problema é inserir todos os cursos em P e retirá-los, um a um. Para cada curso com gratuidade g e conhecimento v retirado, tentamos alocar ele o mais distante possível, para que seja possível, posteriormente, encaixar um curso de mais curta duração. Isso é feito percorrendo um vetor de booleanos, $S[0, m]$, que representa as semanas livres, da posição g até a posição 1. Se existe alguma posição livre (marcada com 0) o curso é encaixado nessa semana (preenche-se um na posição) e soma-se ao total de conhecimento o valor v. Caso contrário, o curso não é incluído.

A complexidade total da solução, no pior caso, é $\Theta(m \cdot n \lg n)$, visto que, para cada elemento retirado, precisamos verificar se existe uma semana livre no vetor de semanas.

Cotas e Rateio Linear

Uma simulação simples utilizando uma fila irá levar tempo $\Theta(Q)$, o que é inviável, visto que Q pode assumir valores de magnitude 10^{18}. Outra abordagem há de ser utilizada.

Com uma fila de prioridades, conseguimos eliminar um participante por rodada. Como N é da ordem 10^5, essa estrutura é mais adequada. A ideia é a seguinte, monta-se uma fila de prioridades P contendo a demanda dos participantes de modo que a maior prioridade seja dado aos menores elementos. Retira-se o valor v da fila de menor prioridade. Para distribuir v cotas aos N participantes, precisaríamos de vN cotas. Se existem cotas suficientes, aquele participante que havia pedido v cotas, sai do jogo, N é diminuído de 1 e o número total de cotas, Q, é ajustado. Caso contrário, distribui-se todas as cotas aos participantes, uma de cada vez, mas sem simular. Isto é, os participantes recebem, cada, $\lfloor \frac{Q}{N} \rfloor$ cotas e sobram $Q \mod N$ cotas. Por fim, essas $Q \mod n$ cotas são distribuídas, na ordem, aos participantes restantes.

A complexidade total do algoritmo é $\Theta(N \lg N) + \Theta(N) = \Theta(N \lg N)$.

Mediana Flutuante

Podemos usar duas filas de prioridade P_1 e P_2 da seguinte forma:

- P_1 dá maior prioridade para os maiores elementos.
- P_2 dá maior prioridade para os menores elementos.

Para cada elemento da entrada, inserimos ele em P_2. Caso $|P_1| = |P_2| - 2$, retiramos o menor elemento de P_2 e o colocamos em P_1, desta forma, P_2 só terá, no máximo, 1 elemento a mais que P_1. Com essa disposição, a mediana será

- Ou o menor elemento de P_2;
- ou o maior elemento de P_1.

Se a quantidade de elementos considerando P_1 e P_2 for par, a mediana estará em P_1. Caso contrário, estará em P_2, logo, ela pode ser consultada em tempo $\Theta(1)$.

A complexidade total da solução é $\Theta(n \lg n)$ em que n é número de elementos, haja vista que inserção e remoção em heaps gastam tempo $\Theta(\lg n)$.