Exercício 1 (*árvore binária de busca*). Uma árvore binária de busca é uma estrutura de dados para representa um conjunto S munido de uma ordem total. Vamos assumir aqui que $S \subset \mathbb{Z}$. Uma árvore binária de busca é uma árvore binária com raiz e tal que a cada nó está associado um inteiro de S de modo que se $v \in S$ está em um nó então todos os nós da subárvore esquerda estão associados a elementos de S menores que v e todos os nós da subárvore direita estão associados a elementos de S maiores que v. A Figura 1 mostra uma árvore binária de busca para $S = \{3, 8, 9, 11, 13, 17, 19\}$. Uma

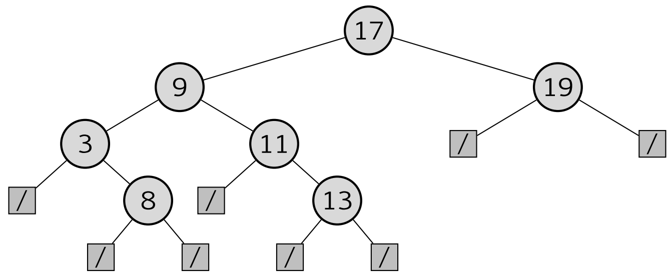

Figura 1: exemplo de uma árvore binária de busca.

busca por x numa árvore binária de busca começa examinando o nó raiz. Se o valor x é igual ao da raiz, a busca termina. Se o valor x é menor do que o da raiz então a busca segue pela subárvore esquerda e se o valor x é maior do que o da raiz, a busca segue pela subárvore direita. Esse processo é repetido até o valor ser encontrado ou a subárvore ser vazia. O tempo de busca é, no pior caso, proporcional à altura da árvore, isto é, o maior número de arestas percorridas num caminho da raiz até alguma folha. No exemplo da Figura 1 a altura é 3.

A inserção de x numa árvore binária de busca começa com uma busca até chegar numa subárvore vazia, é nesse local que o elemento é inserido. Por exemplo, a inserção de 2, 15 e 20 na árvore mostrada na Figura 1 resulta na árvore da Figura 2 abaixo.

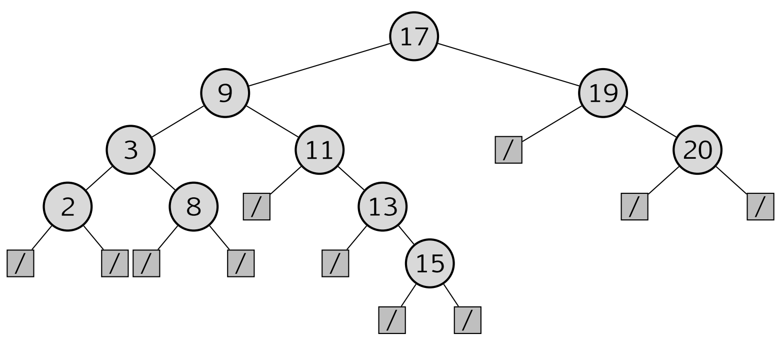

Figura 2: árvore de busca binária obtida a partir do exemplo acima após a inserção de 2, 15 e 20.

A árvore da Figura 2 pode ser obtida a partir da árvore vazia e inserindo-se os elementos de S, um a um, na seguinte ordem: $17, 9, 3, 2, 8, 11, 13, 15, 19, 20$.

1. Qual seria a árvore resultante se os elementos fossem inseridos na ordem: $2, 3, 8, 9, 11, 13, 15, 17, 19, 20$?

2. Seja $S \subset \mathbb{Z}$ finito, não vazio e de cardinalidade n. Prove que se uma árvore de busca binária é construída tomando-se uma permutação aleatória dos elementos de S e inserindo-se os elementos na ordem definida pela permutação, então a altura da árvore obtida é $O(\log n)$ com probabilidade $1 - n^{-c}$, para alguma constante $c > 0$. Conclua que com alta probabilidade a árvore é construída em tempo $O(n \log n)$.

Exercício 2. Prove que $\mathrm{Var}[X] = 0$ então $\mathbb{P}[X = \mathbb{E}\, X] = 1$.

Exercício 3 (propriedades da covariância). Sejam X e Y variáveis aleatórias quadrado integráveis, $a, b \in \mathbb{R}$. Verifique que valem

1. $\mathrm{Cov}(X, X) = \mathrm{Var}[X]$;
2. $\mathrm{Cov}(X, Y + a) = \mathrm{Cov}(X, Y)$;
3. $\mathrm{Cov}(X, Y) = \mathrm{Cov}(Y, X)$;
4. $\mathrm{Cov}(X, a \cdot Y + b \cdot Z) = a \cdot \mathrm{Cov}(X, Y) + b \cdot \mathrm{Cov}(X, Z)$;
5. $|\mathrm{Cov}(X, Y)| \leqslant \sqrt{\mathrm{Var}[X]}\sqrt{\mathrm{Var}[Y]}$.

Exercício 4. Prove que se $X_1, \dots, X_n$ são variáveis aleatórias independentes com distribuição de Rademacher, $t \in (0,1)$ e $s > 0$, então

$$\mathbb{P}\left[\frac{1}{n}\sum_{i=1}^{n} X_i \geqslant t\right] \leqslant \exp\left(-\frac{n}{2}((1+t)\ln(1+t) + (1-t)\ln(1-t))\right).$$

(*dica*: minimize $f(s) = \cosh(s)\exp{-st}$ usando as técnicas de Cálculo Diferencial.)

Exercício 5. Suponha que $\mathcal{H}$ seja uma família 2-universal de funções $\{0,1\}^n \to \{0,1\}^n$. Para $m < n$ podemos definir uma família $\mathcal{H}'$ de funções $\{0,1\}^n \to \{0,1\}^m$ fazendo para cada $h \in \mathcal{H}$

$$h'(x) := h(x) \restriction_m$$

em que $h(x) \restriction_m$ denota a restrição de $h(x)$ aos primeiros m bits. $\mathcal{H}'$ é uma família 2-universal?

Exercício 6. Uma família de funções $\mathcal{H} = \{ h_\lambda \in N^U : \lambda \in \Lambda \}$ tais que para quaisquer $x \neq y$

$$\mathop{\mathbb{P}}_{\lambda \in \Lambda}[h_\lambda(x) = h_\lambda(y)] \leqslant \frac{1}{|N|}.$$

é chamada de *hash fracamente 2-universal*. Verifique que a família de funções dada no Exemplo 3.18 (pág. 19 do cap3.pdf – notas da semana 4) não é 2-universal, mas é fracamente 2-universal.

Exercício 7 (*desaleatorização de corte grande*). Seja $G = (V, E)$ uma grafo sobre $V = \{0, 2, \dots, n-1\}$. Uma função $h \colon V \to \{0,1\}$ determina o subconjunto $A \subset V$ dado por $A = \{ i \in V : h(i) = 1 \}$ o qual, por sua vez, determina o corte $\nabla(A)$. Prove que se se sorteamos h em uma família de funções de *hash* fracamente 2-universal então $\mathbb{E}\,|\nabla(A)| \geqslant |E|/2$. Escreva e analise um algoritmo aleatorizado baseado nessa ideia. Escreva e analise um algoritmo (determinístico) que *desaleatoriza* o algoritmo que você escreveu usando a família fracamente 2-universal do Exemplo 3.18.

Exercício 8. Dizemos que X é uma (ε, δ)-aproximação para a quantidade V se $\mathbb{P}[|X - V| \leqslant \varepsilon V] \geqslant 1 - \delta$. Sejam $X_1, \dots, X_m$ variáveis aleatórias independentes e com mesma distribuição sobre $\{0,1\}$ e $\mathbb{E}\, X_i = \mu$ para todo i. Denotemos por $\overline{X}_m$ a média amostral, $\overline{X}_m = (1/m)\sum_{i=1}^{m} X_i$. Prove que se $m \geqslant 3\ln(2/\delta)/(\varepsilon^2\mu)$ então

$$\mathbb{P}[\,|\overline{X}_m - \mu| \geqslant \varepsilon\mu\,] \leqslant \delta$$

e disso segue que a média amostral é uma (ε, δ)-aproximação para μ.

Exercício 9. Suponha que sejam dados conjuntos $S_1, S_2, \dots, S_n$ com cardinalidades conhecidas. O algoritmo a seguir estima a cardinalidade da união desses conjuntos. Defina $\mathrm{cov}(x) := |\{i : x \in S_i\}|$ e $s := \sum_{i=1}^{n} |S_i|$.

Instância: conjuntos $S_1, S_2, \dots, S_n$.
Resposta : uma estimativa para $S_1 \cup S_2 \cup \dots \cup S_n$.

escolha $S \in_{\mathcal{D}} \{S_1, S_2, \dots, S_n\}$ de acordo com lei $\mathcal{D}(S_i) = |S_i|/s$
$x \leftarrow_R S$
compute $\mathrm{cov}(x)$
$X \leftarrow \frac{s}{\mathrm{cov}(x)}$
responda X.

Algoritmo 1: Algoritmo de Karp–Luby

Prove que $\mathbb{E}\, X = \left|\bigcup_{i=1}^{n} S_i\right|$ e que $\mathrm{Var}[X] \leqslant \mathbb{E}[X]^2$.

Exercício 10. Sejam $X_1, X_2, \dots, X_m$ as respostas de m rodadas independentes do Algoritmo 1 com a mesma entrada e $\overline{X}_m$ a média amostral dessas rodadas. Prove que se $m \geqslant (9/2)(n-1)(1/\varepsilon^2)\ln(2/\delta)$ então temos uma (ε,δ)-aproximação para $\left|\bigcup_{i=1}^n S_i\right|$.

Exercício 11. Uma fórmula DNF é uma fórmula booleana com n variáveis e m cláusulas disjuntivas $C_1, \dots, C_m$; uma cláusula é uma disjunção se for um "e" de literais, a fórmula é um "ou" dessas cláusulas. O problema #DNF, ou o problema de contagem DNF, toma como entrada uma fórmula DNF e retorna o número de valorações que satisfazem a fórmula.

Escreva um algoritmo que, para dados $\varepsilon > 0$, $\delta > 0$ e φ fórmula DNF, obtenha uma (ε,δ)-aproximação para estimar o número de valorações que satisfazem uma fórmula DNF em $O(m\varepsilon^{-2}\ln(1/\delta))$ rodadas e tempo de execução total $O(nm^2\varepsilon^{-2})\ln(1/\delta))$.

Exercício 12. Seja $G = (V,E)$ um grafo sobre $V = \{1,2,\dots,n\}$. Assuma que G tenha as seguintes propriedades: (1) todo vértice tem grau d; (2) para todo $W \subseteq V$ com $|W| \leqslant n/2$ vale que $|N(W)| \geqslant c|W|$, onde $N(W) = \{x \in V \colon \{x,w\} \in E \text{ para algum } w \in W\}$. Um *passeio aleatório* em G é uma sequência $(X_i \colon i \in \mathbb{N})$ de variáveis aleatórias com $X_0 \in_R V$ e $X_{i+1} \in_R N(X_i)$.

Prove que se $B \subset V$ é tal que $|B| = \beta n$ para alguma constante $\beta > 0$, então para todo $\delta > 0$

$$\mathbb{P}\left[\left|\frac{1}{k}\sum_{i=1}^{k}\mathbb{1}_{[X_i \in B]} - \beta\right| > \delta\right] < 2\exp\left(-\frac{(1-c)\delta^2 k}{4}\right).$$