

Grafos Aleatórios

Seja $G(n)$ a coleção de todos os grafos com conjunto de vértices $V := \{1, \dots, n\}$. Qualquer propriedade invariante sob isomorfismo define uma subcoleção de $G(n)$. Assim, convém confundir os conceitos de propriedade e subcoleção de $G(n)$. Diremos que quase todo grafo tem determinada propriedade $P(n)$ se

Uma forma de se estudar o conjunto $G(n)$ é baseada na introdução de uma medida de probabilidade nesse conjunto. Seja p um número no intervalo $(0, 1)$ e escolha cada elemento $V^{(2)}$, independente, com probabilidade p . Se A é o conjunto dos pares escolhidos, então (V, A) é um grafo aleatório em $G(n)$. A probabilidade de que o grafo (V, A) assim construído seja idêntico a um determinado elemento de $G(n)$ que tenha m arestas é

$$p^m(1-p)^{N-m}.$$

Se — então todos os $2N$ grafos em $G(n)$ são equiprováveis: a probabilidade de obter qualquer um deles é —.

As funções abaixo constroem grafos aleatórios com vértices $0..V-1$.

Um gerador de grafos aleatórios

A primeira função gera grafos aleatórios com exatamente E arestas.

```
Graph GRAPHrand1 (int V, int E) {
  Graph G = GRAPHinit (V);
  while (G->A < 2*E) {
    Vertex v = randV(G);
    Vertex w = randV(G);
    GRAPHinsertE(G, v, w);
  }
  return G;
}
```

A função acima gera grafo aleatório com vértices $0..V-1$ e exatamente E arestas.

A função `GRAPHrand1` só deve ser invocada com E bem menor que $V * (V - 1) / 2$. À medida que E se aproxima desse limite, a execução da função consumirá cada vez mais tempo.

A função abaixo devolve um vértice aleatório do grafo G .

```
Vertex randV(Graph G) {
  return G->V * (rand() / (RAND_MAX + 1.0));
}
```

O código pode ser substituído para $G->V \leq \text{RAND_MAX}+1$.

Para ter acesso à função `rand` e à constante `RAND_MAX` você precisa do arquivo-interface `stdlib.h`.

A função `int rand (void)` devolve um número inteiro aleatório entre 0 e `RAND_MAX` inclusive, ou seja, um número inteiro no intervalo fechado $0..RAND_MAX$.
`RAND_MAX = 32767.`

A função `rand` (abreviatura de *random*), definida na biblioteca `stdlib`, gera números aleatórios. Cada invocação da função produz um número aleatório no intervalo fechado $0..RAND_MAX$.

Graus de vértices

O grau de um vértice em um grafo é o número de arestas que incidem no vértice. Esse número é igual ao grau de entrada do vértice e também ao grau de saída do vértice.

A soma dos graus de todos os vértices de um grafo vale $2E$, sendo E o número de arestas. O grau médio do grafo é o número $2E/V$, sendo V o número de vértices.

Um vértice é isolado se o seu grau é nulo.

Número máximo de arestas

Um grafo com V vértices tem $V \times (V - 1) / 2$ arestas.

Um grafo é completo se todo par não ordenado de vértices distintos é uma aresta. Um grafo completo com v vértices tem exatamente $V \times (V - 1) / 2$ arestas.

Outro gerador de grafos aleatórios

A função abaixo gera grafos aleatórios que têm E arestas em média.

```
Graph GRAPHrand2 (int V, int E) {
    Vertex v, w;
    double p = 2.0 * E / V / (V - 1);
    Graph G = GRAPHinit(V);
    for (v = 0; v < V; v++)
        for (w = 0; w < v; w++)
            if (v != w && rand() <= p * RAND_MAX)
                GRAPHinsertE(G, v, w);
    return G;
}
```

A função `GRAPHrand2` gera um grafo aleatório com vértices $0..V-1$. Cada uma das $V \times (V-1) / 2$ possíveis arestas é gerada com probabilidade p , sendo p calculado de modo que o número esperado de arestas seja E (desde que $E \leq V \times (V-1) / 2$). A função supõe que $V \geq 2$.

Exercícios

1. O seguinte programa promete simular uma jogada de um dado de 6 faces. Qual o defeito do programa?

```
int RolaDado (void) {
    int r;
    r = rand ();
    if (r < RAND_MAX / 6) return 1;
    else if (r < RAND_MAX * 2 / 6) return 2;
    else if (r < RAND_MAX * 3 / 6) return 3;
    else if (r < RAND_MAX * 4 / 6) return 4;
    else if (r < RAND_MAX * 5 / 6) return 5;
    else return 6;
}
```

2. O seguinte programa promete simular uma jogada de uma moeda. Qual o defeito do programa?

```
char *RolaMoeda (void) {
```

```

int r;
r = rand () % 2;
if (r == 1) return "cara";
else return "coroa";
}

```

3. O que acontece se a função GRAPHrand1 for chamada com $E > V * (V - 1) / 2$?
4. O que acontece se a função GRAPHrand2 for chamada com $E > V * (V - 1) / 2$?
5. Escreva uma função que receba inteiros n e E e gera um subgrafo aleatório de uma grade quadrada com $n \times n$ vértices. Cada uma das arestas da grade deve ser incluída no grafo com probabilidade p , sendo p calculado de modo que o número esperado de arestas seja E .
6. Use a função DIGRAPHshow para exibir o grafo gerado.

Bibliografia

Estrutura de Dados Usando C

Aaron M. Tenenbaum, Yedidyah Langsam, Moshe J. Augenstein
São Paulo: Makron Books, 1995

Algoritmos: Teoria e Prática

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
Rio de Janeiro: Elsevier, 2002

Estrutura de Dados

Paulo Veloso, Clesio dos Santos, Paulo Azevedo, Antonio Furtado
Rio de Janeiro: Campus, 1983

Fundamentos de Estrutura de Dados 253

Ellis Horowitz, Sartaj Sahni
Rio de Janeiro: Campus, 1986

Uma Introdução Sucinta à Teoria dos Grafos 8

<http://www.ime.usp.br/~pf/teoriadosgrafos/>
P. Feofiloff, Y. Kohayakawa, Y. Wakabayashi

Algoritmos para Grafos em C via *Sedgewick*

http://www.ime.usp.br/~pf/algoritmos_para_grafos/
Paulo Feofiloff