Grids

Grades retangulares, triangulares, hexagonais, ...

Profs. André, Salles

Departamento de Informática
Universidade Federal de Viçosa

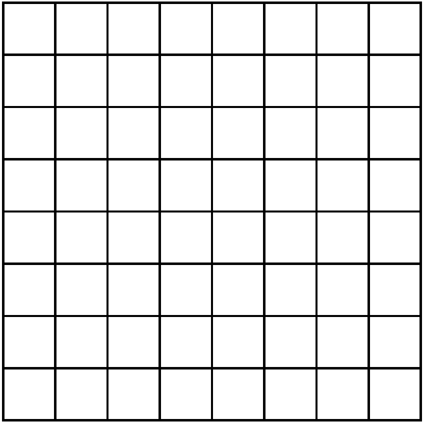
INF 333 - 2024/1

[retirado do capítulo 11 do livro Programming Challenges]

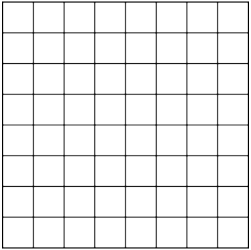
Intro

> It is not that polar coordinates are complicated, it is just that Cartesian coordinates are simpler than they have a right to be.
>
> ---
>
> *Kleppner and Kolenhow*
> *"An Introduction to Mechanics"*

Grade quadrada (regular, cartesiana)



Grade quadrada



- Vértices
- Arestas
- Células

- Cada vértice toca 4 arestas e 4 células (exceto os da borda)
- Cada célula toca outras 8 células (sendo 4 diagonalmente)
- Numa grade 3D, cada vértice toca 6 arestas e 8 células
- Cada célula toca outras 26 células
- Em d dimensões, cada vértice toca $2d$ arestas e 2^d células

Grade quadrada - representação

- Representação natural: matriz
- Célula `[i][j]` representa célula (i, j) da grade
- 4 células vizinhas: somar ± 1 a uma das coordenadas
- 8 células vizinhas: somar ± 1 a uma ou às duas coordenadas
- Note que a matriz pode representar os vértices ou as células
  - De fato, o dual[1] de uma grade quadrada é uma grade quadrada um pouco menor

[1] no dual cada célula vira um vértice, conectado aos vértices que representam células adjacentes por arestas

Grade quadrada - caminhamento

Por linha

```
(1,1)    row_major(int n, int m)
(1,2)    {
(1,3)        int i,j;        /* counters */
(2,1)
(2,2)        for (i=1; i<=n; i++)
(2,3)            for (j=1; j<=m; j++)
(3,1)                process(i,j);
(3,2)    }
(3,3)
```

Grade quadrada - caminhamento

Por coluna

```
(1,1)    column_major(int n, int m)
(2,1)    {
(3,1)        int i,j;        /* counters */
(1,2)
(2,2)        for (j=1; j<=m; j++)
(3,2)            for (i=1; i<=n; i++)
(1,3)                process(i,j);
(2,3)    }
(3,3)
```

Grade quadrada - caminhamento

Snake order

```
(1,1)   snake_order(int n, int m)
(1,2)   {
(1,3)     int i,j;          /* counters */
(2,3)
(2,2)     for (i=1; i<=n; i++)
(2,1)       for (j=1; j<=m; j++)
(3,1)         process(i, j + (m+1-2*j) * ((i+1) % 2));
(3,2)   }
(3,3)
```

- linha por linha, alternando a direção
- por exemplo, impressora matricial imprimindo na “ìda” e na “volta”

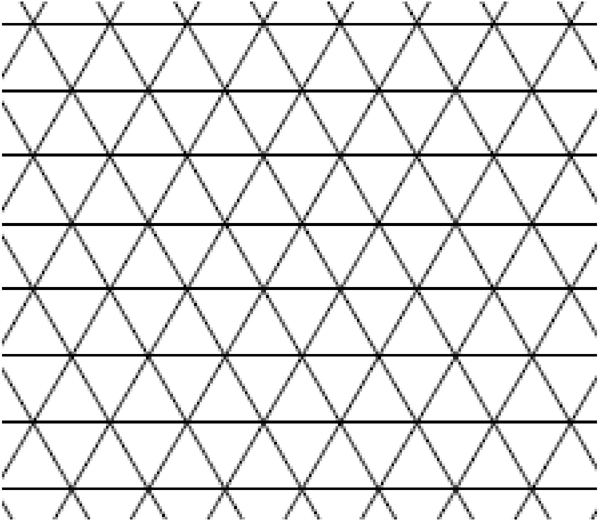
Grade quadrada - caminhamento

Diagonal order

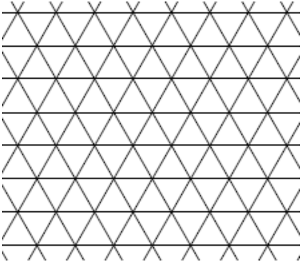
```
(1,1)  diagonal_order(int n, int m)
(2,1)  {
(1,2)     int d,j;      /* diagonal and point counters *
(3,1)     int pcount;   /* points on diagonal */
(2,2)     int height;   /* row of lowest point */
(1,3)
(4,1)     for (d=1; d<=(m+n-1); d++) {
(3,2)        height = 1 + max(0,d-m);
(2,3)        pcount = min(d, (n-height+1));
(4,2)        for (j=0; j<pcount; j++)
(3,3)           process(min(m,d)-j, height+j);
(4,3)     }
       }
```

- são $m + n - 1$ diagonais, com número variável de elementos

Grade triangular

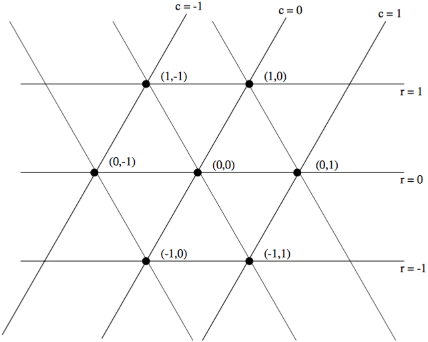


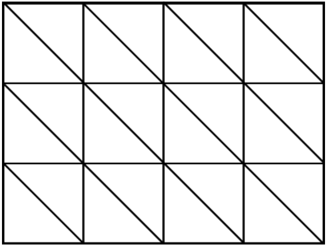
Grade triangular



- Construída por 3 conjuntos de linhas igualmente espaçadas
  - linhas horizontais
  - “colunas” formando 60° com as linhas
  - “diagonais” formando 120° com as linhas
- Vértice é a interseção de 3 linhas
- As células (faces) são triângulos equiláteros
- Cada vértice toca 6 arestas; é conectado a 6 outros vértices

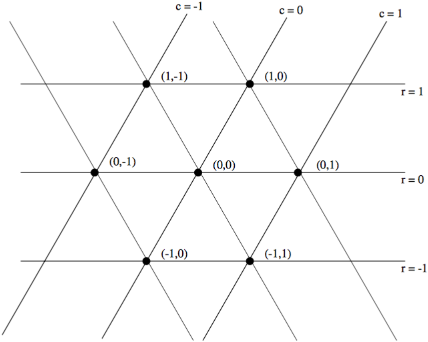
Grade triangular - representação





- Embora um vértice seja definido por 3 linhas,
  2 dimensões são suficientes: por exemplo, linha e “coluna”
- Um vértice é escolhido como origem $(0, 0)$
- Os demais estão *x* linhas acima e *y* colunas de 60° à direita
- Vizinhos no sentido anti-horário: somar
  $(0, 1), (1, 0), (1, -1), (0, -1), (-1, 0), (-1, 1)$ às coordenadas.

Grade triangular - coordenadas geométricas

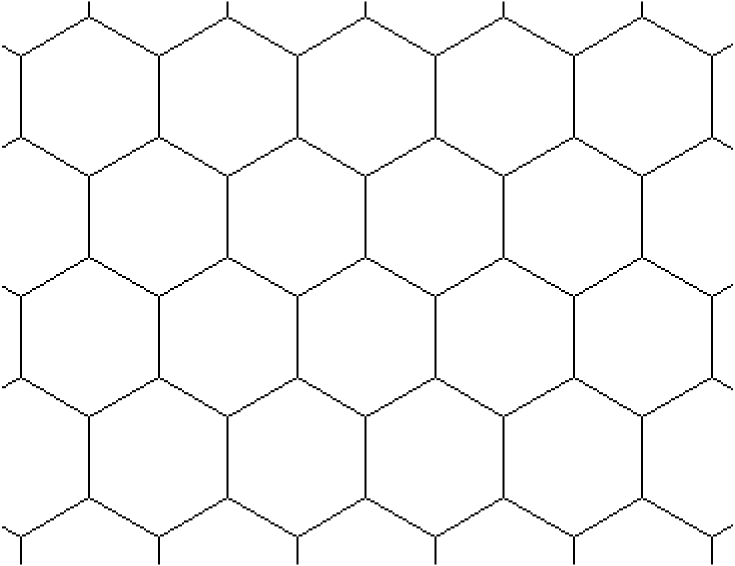


- Note que os vértices ocorrem em coordenadas x de $\frac{1}{2}$ em $\frac{1}{2}$
- Seja d a distância entre um vértice e cada um de seus 6 vizinhos
- Por trigonometria mostra-se que (x_t, y_t) está no ponto geométrico

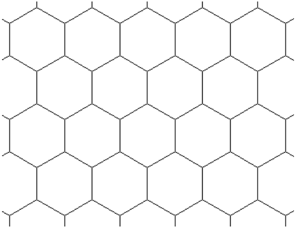
$$(x_g, y_g) = (d(x_t + (y_t \cos(60^\circ))), dy_t \sin(60^\circ))$$

- Por eficiência, substitui-se $\cos 60^\circ = \frac{1}{2}$ e $\sin 60^\circ = \frac{\sqrt{3}}{2}$

Grade hexagonal

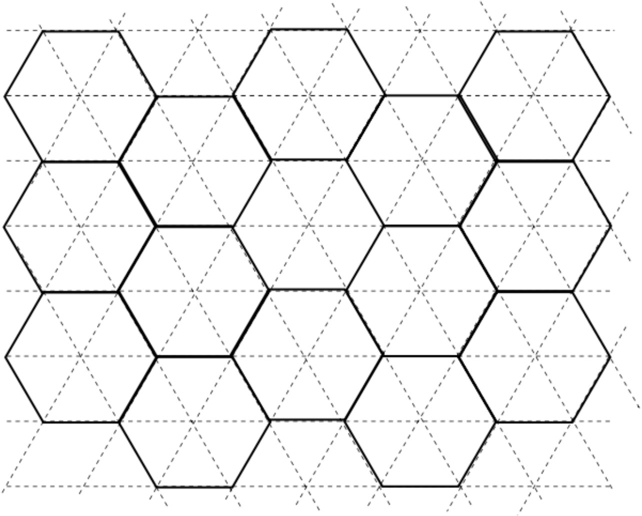


Grade hexagonal



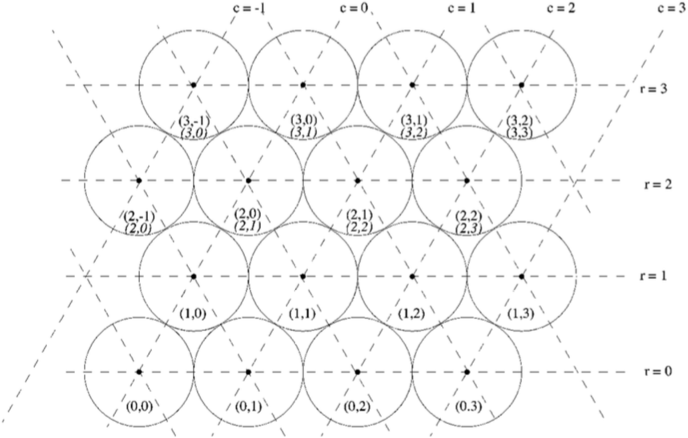
- Cada face é um hexágono, que é adjacente a outros 6
- Vértices tocam 3 arestas; são vizinhos de outros 3
- Propriedades interessantes:
  - hexágonos são mais “redondos” que quadrados; e círculos tem mais área por perímetro, mais eficientes para construir
  - grades hexagonais são mais rígidas; por isso abelhas usam hexágonos...

Grade hexagonal - representação



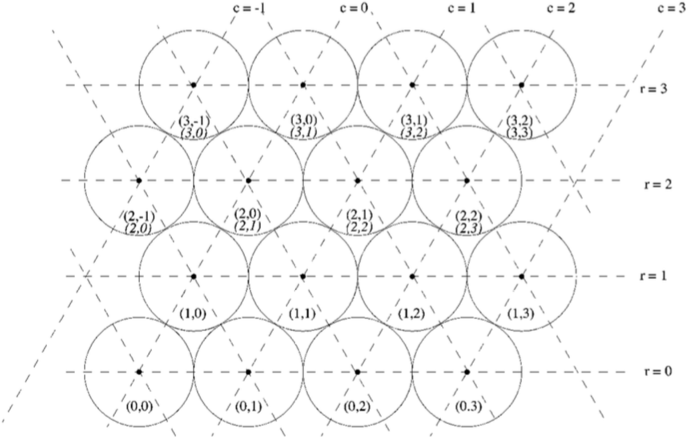
- apagar vértice sim e dois não de uma triangular: vira hexagonal!
- na verdade, uma é o dual da outra

Grade hexagonal - representação



- representação de hexágonos como feito pros vértices da triangular
- hexágonos adjacentes somando-se (0,1),(1,0),(1,-1),(0,-1),(-1,0),(-1,1)
- mas fica em forma de losango...
- uma correção para formar um retângulo é mostrada em itálico

Grade hexagonal - representação




```
hex_to_array(int xh, int yh, int *xa, int *ya)
{
        *xa = xh;
        *ya = yh + xh - ceil(xh/2.0);
}
```

Grade hexagonal - representação

De matriz para hexagonal

```
array_to_hex(int xa, int ya, int *xh, int *yh)
{
        *xh = xa;
        *yh = ya - xa + ceil(xa/2.0);
}
```

De hexagonal para matriz

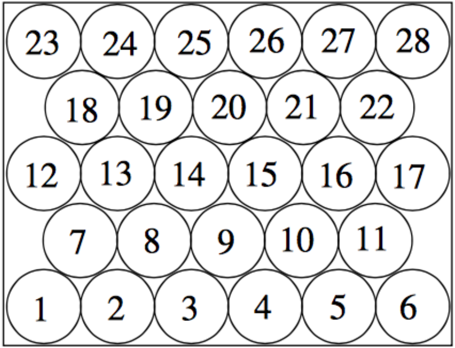
```
hex_to_array(int xh, int yh, int *xa, int *ya)
{
        *xa = xh;
        *ya = yh + xh - ceil(xh/2.0);
}
```

Grade hexagonal - coordenadas geométricas

- origem é o centro de um disco em $(0, 0)$
- coordenada hexagonal (x_h, y_h) é o centro de um disco na horizontal x_h “vertical” y_h
- como na triangular, mas disco de raio r, metade da distância d

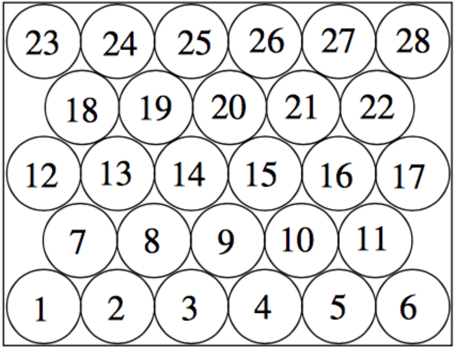
```
hex_to_geo(int xh, int yh, double r, double *xg, double *yg)
{
    *yg = (2.0 * r) * xh * (sqrt(3)/2.0);
    *xg = (2.0 * r) * xh * (1.0/2.0) + (2.0 * r) * yh;
}

geo_to_hex(double xg, double yg, double r, double *xh, double *yh)
{
    *xh = (2.0/sqrt(3)) * yg / (2.0 * r);
    *yh = (xg - (2.0 * r) * (*xh) * (1.0/2.0) ) / (2.0 * r);
}
```

Grade hexagonal - Plates in a box



- Caixa de tamanho $L \times W$, pratos de raio r
- Camada inferior comporta $p = \lfloor W/(2r) \rfloor$ pratos
- Demais contém p ou alternam com $p - 1$ e p pratos (depende da relação W e r)

Grade hexagonal - Plates in a box



- Quantos pratos de diâmetro unitário cabem numa grade 4×4 de leiaute hexagonal? 14; no leiaute retangular caberiam 16...
- Mas numa 10×10 cabem 105! (contra 100)
- Numa 100×100 caberiam 11443... vantagem significativa.

Exemplo: UVa 10161 - Ant on a Cheesboard

Resumo

Uma formiga anda numa grade $M \times M$ seguindo o padrão ao lado, que mostra onde ela está no passo $1, 2, 3, \ldots$
Ela continua repetindo: sobe, direita, desce, direita, sobe, esquerda, ...

Dado o número do passo, determinar a posição da formiga.

25	24	23	22	21
10	11	12	13	20
9	8	7	14	19
2	3	6	15	18
1	4	5	16	17

Exemplos

Entrada

```
8
20
25
0
```

Saída

```
2 3
5 4
1 5
```

Exemplo: UVa 10182 - Bee Maja

Resumo

Bee é uma abelha que vive numa colmeia hexagonal com milhares de outras. Seu amigo Willi marcou de encontrá-la. O problema é que eles usam diferentes sistemas de coordenadas... enquanto ela usa um sofisticado sistema 2D para se movimentar, ele, menos inteligente, fica dando voltas no sentido horário a partir do meio da colmeia.

Ajude Maja a se orientar no sistema de Willi.

Dado o número de um favo no sistema de Willi, informe a coordenada no sistema de Maja.