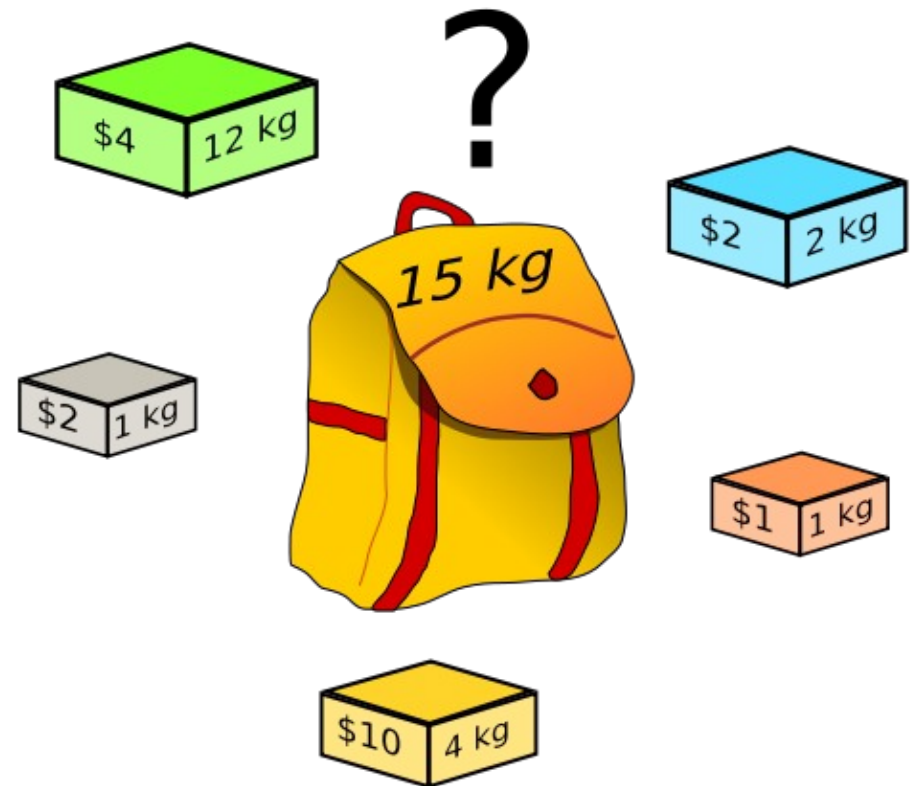


025089 – Projeto e Análise de Algoritmos

Aula 08

Exemplo 2

- Problema da mochila:
 - Dado um conjunto de itens, cada um com seu peso w e valor v definidos, encontrar o(os) subconjunto(s) de maior valor que cabe em uma mochila de tamanho K

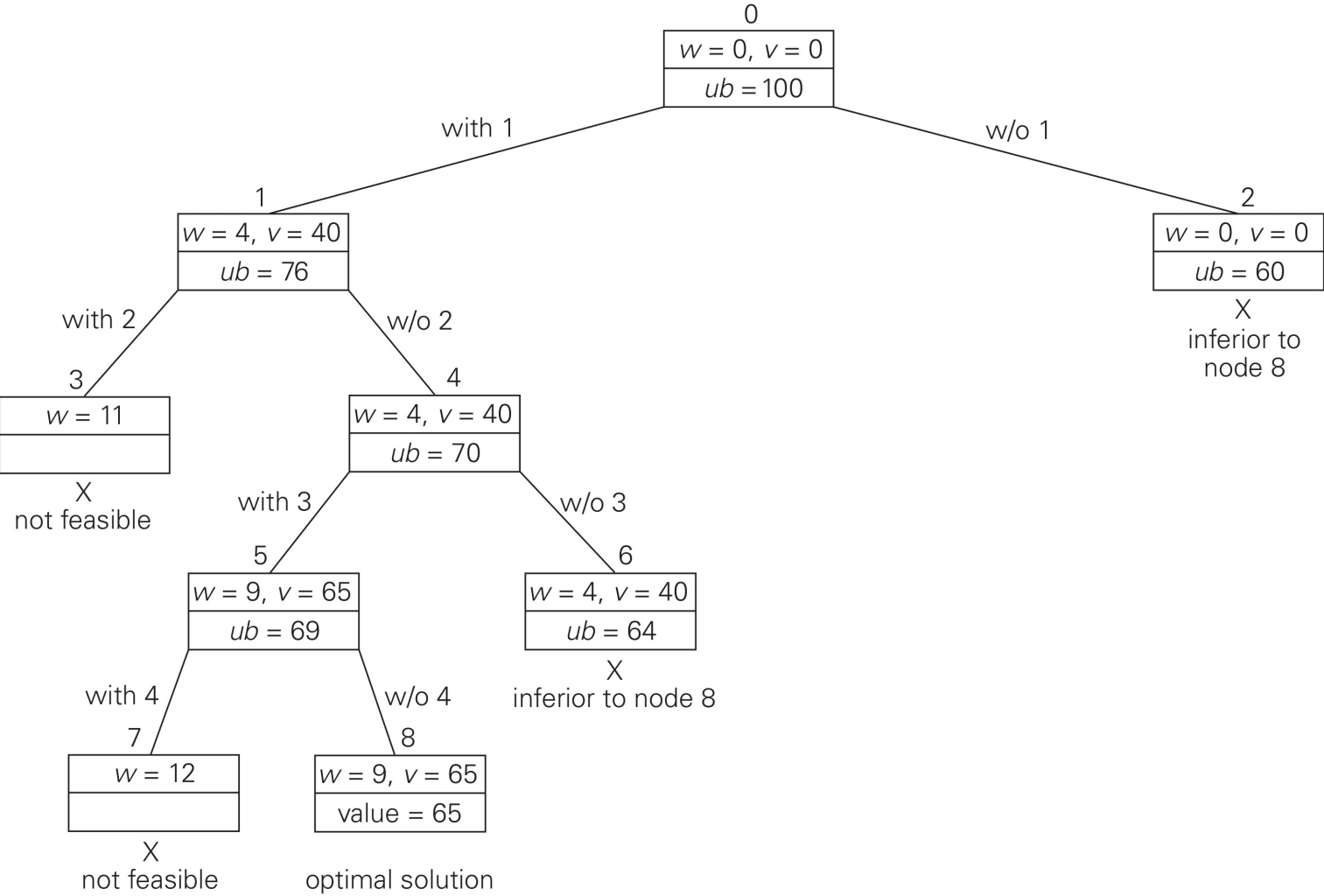


Exemplo 2

- Algoritmo (força-bruta):
 - Gerar todas as possibilidades (quantas?)
 - Subconjuntos que não cabem na mochila não são possibilidades
 - Calcular o custo de cada uma
 - Escolher o subconjunto de menor custo
- Backtracking
 - Árvore binária, inclui ou não o item na mochila

item	weight	value	<u>value</u> <u>weight</u>
1	4	\$40	10
2	7	\$42	6
3	5	\$25	5
4	3	\$12	4

The knapsack's capacity W is 10.



Exemplo 2

```
// int K, int N, int V[], int W[]
// int cmax, int imax, int atu[], int sol[]

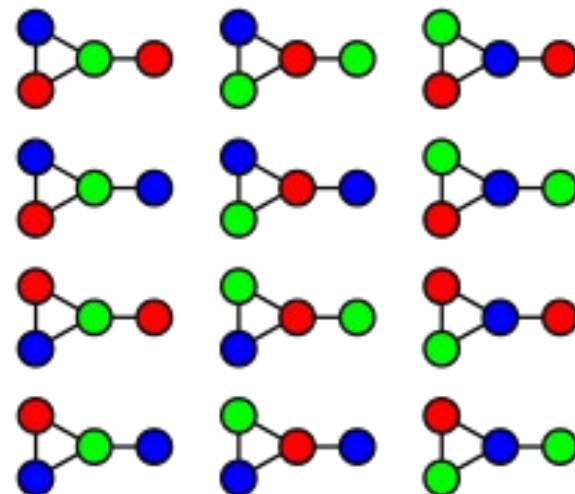
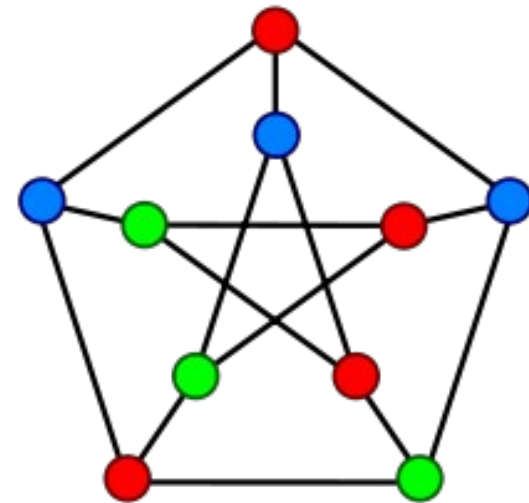
void copySolution(int n) {
    for( int j=0; j<n; j++)
        sol[j] = atu[j];
    imax = n;
}

void ksp( int i, int sel_n, int sel_w, int sel_c ) {
    if (sel_w <= K && sel_c > cmax ) {
        cmax = sel_c;
        copySolution(sel_n);
    }

    if( sel_w < K && i<N ) {
        ksp(i+1,sel_n,sel_w,sel_c);
        atu[sel_n] = i;
        ksp(i+1,sel_n+1,sel_w+W[i], sel_c+V[i]);
    }
}
```

Exemplo 4

- Colorir os vértices de um grafo com no máximo ***K*** cores, não permitindo vértices adjacentes com cores iguais



Exemplo 4

- Algoritmo (força-bruta):
 - Gerar todas as possibilidades (quantas?)
 - Verificar se vértices vizinhos possuem a mesma cor
- Backtracking
 - Colorir um vértice com a primeira cor disponível
 - Processar os vizinhos
 - Se não retornar solução, trocar de cor

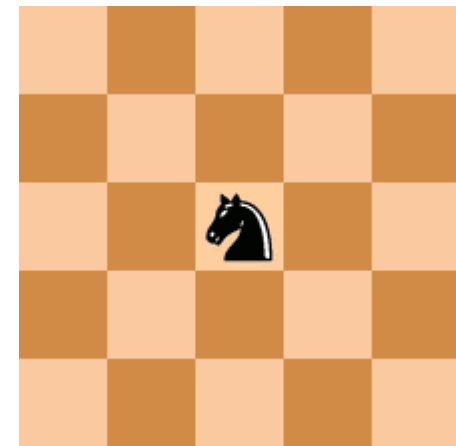
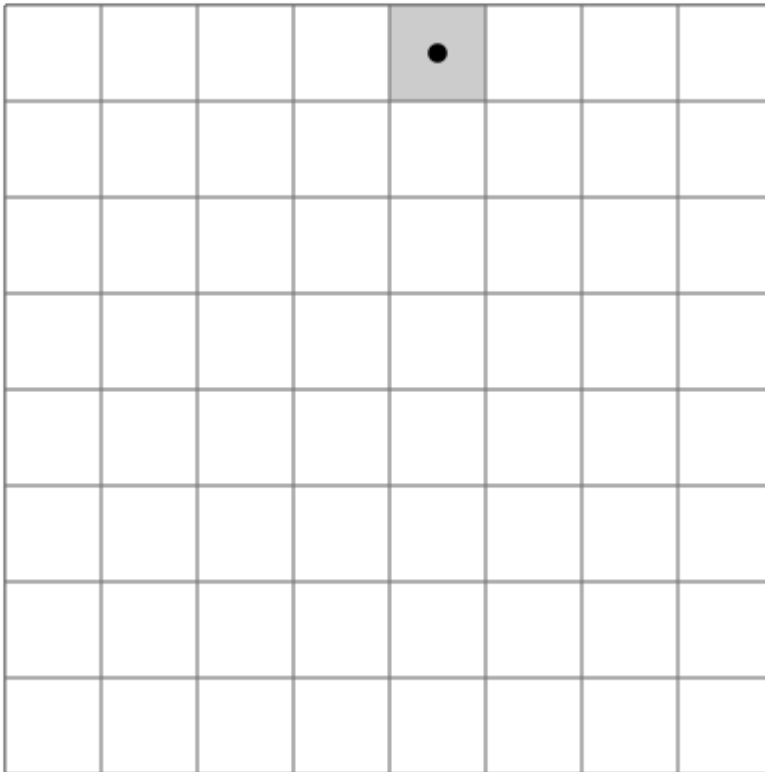
Exemplo 4

```
bool isSafe (int v, bool graph[V][V], int color[], int c) {
    for (int i = 0; i < V; i++)
        if (graph[v][i] && c == color[i])
            return false;
    return true;
}

bool gc(bool graph[V][V], int m, int color[], int v) {
    if (v == V)
        return true;
    for (int c = 1; c <= m; c++) {
        if (isSafe(v, graph, color, c)) {
            color[v] = c;
            if (gc(graph, m, color, v+1) == true)
                return true;
            color[v] = 0;
        }
    }
    return false;
}
```


Exemplo 5

- Knight's Tour
 - A partir de uma posição, encontrar um caminho que percorra todas as posições do tabuleiro



Exemplo 5

- Backtracking
 - Escolher um movimento
 - Recursão
 - Se não encontrou resposta, trocar o movimento

Exemplo 5

```
int isSafe(int x, int y, int sol[N][N]) {
    if ( x >= 0 && x < N && y >= 0 && y < N && sol[x][y] == -1)
        return 1;
    return 0;
}

int KT(int x,int y,int movei,int sol[N][N],int xMove[N],int yMove[N]){
    int k, next_x, next_y;
    if (movei == N*N)
        return true;
    for (k = 0; k < 8; k++) {
        next_x = x + xMove[k];
        next_y = y + yMove[k];
        if (isSafe(next_x, next_y, sol)) {
            sol[next_x][next_y] = movei;
            if (kt(next_x, next_y, movei+1, sol, xMove, yMove) == true)
                return true;
            else
                sol[next_x][next_y] = -1;// backtracking
        }
    }
    return false;
}
```

