MC-102 — Aula 18
Argumentos para main() e busca exaustiva

Instituto de Computação – Unicamp

Segundo Semestre de 2007

Roteiro

1. Parâmetros para main()
2. Busca exaustiva
3. Problemas clássicos com busca exaustiva

Parâmetros para a função main()

- A passagem de parâmetros à função main() pode ser feita através da linha de comando através das variáveis argc e argv.

Argumentos para a função main()

int main(int argc, char *argv[])

- A variável argc (argument count) contém o número de parâmetros passados através da linha de comando, incluindo o nome do programa.
- A variável argv (argument values) é um apontador para vetores de caracteres (strings). Cada string é um argumento da linha de comando.

Argumentos para a função main()

int main(int argc, char *argv[])
Exemplo: ./*programa* 1 2
argc = 3
argv[0] = string "./programa"
argv[1] = string "1"
argv[2] = string "2"

Busca exaustiva

- Uma das técnicas de resolução de problemas é gerar todas as possíveis soluções de um problema e verificar qual delas é de fato a solução procurada.
- Essa técnica é chamada de busca exaustiva, pois percorremos todo o espaço de possíveis soluções em busca da solução do problema.

Busca exaustiva

- Tipicamente uma solução por busca exaustiva é composta de duas funções: uma que gera todas as possíveis soluções e outra que verifica se a solução gerada é a solução que atende ao problema.
- Um dos problemas com a busca exaustiva é que pode existir um número muito grande de soluções a serem verificadas.

Busca pelo elemento máximo de um vetor

- Podemos modelar a busca pelo maior elemento de um vetor como uma busca exaustiva, da seguinte forma:

  Soluções possíveis: cada elemento do vetor

  Verificação: chamaremos de `MAX` a melhor solução encontrada até o momento e de `X` a solução que está sendo verificada no momento. Se X for igual a MAX, então MAX passa a valer X.

Veja o exemplo em maximo.c

Subconjuntos de um conjunto

Descrição do problema

Dado um conjunto S, cujos elementos estão representados em um vetor, determine todos os possíveis subconjuntos de S (incluindo o subconjunto vazio e o próprio S).

Subconjuntos de um conjunto

- Podemos modelar o problema como uma busca exaustiva, da seguinte forma:

  Soluções possíveis: todos os subconjuntos de S

  Verificação: Imprimir o subconjunto gerado (todos são soluções válidas)

Como gerar os subconjuntos ?

Subconjuntos de um conjunto

- Considere um conjunto P contendo n elementos e considere um elemento qualquer x. Para obtermos todos os subconjuntos de P, devemos obter
  - Obter todos os subconjuntos de P que não contém x. Para fazer isso, basta obter todos os subconjuntos de $P - \{x\}$.
  - Obter todos os subconjuntos de P que contém x. Para fazer isso, basta obter todos os subconjuntos de $P - \{x\}$ e acrescentar x a cada um deles.
- O caso base é quando temos que obter todos os subconjuntos do conjunto vazio: o próprio conjunto vazio.

Veja exemplo em subconjunto.c

Permutações de n elementos

Descrição do problema

Dado um conjunto S, cujos elementos estão representados em um vetor, determine todos os possíveis permutações dos elementos de S.

Permutações de n elementos

- Considere um conjunto P contendo n elementos. Para cada elemento x de P:
  - Colocar o elemento x no começo de nossa permutação.
  - Obter todas as permutações possíveis do conjunto $P - \{x\}$
- O caso base é quando temos que obter as permutações do conjunto vazio: somente ele mesmo.

Veja exemplo em permutacao.c