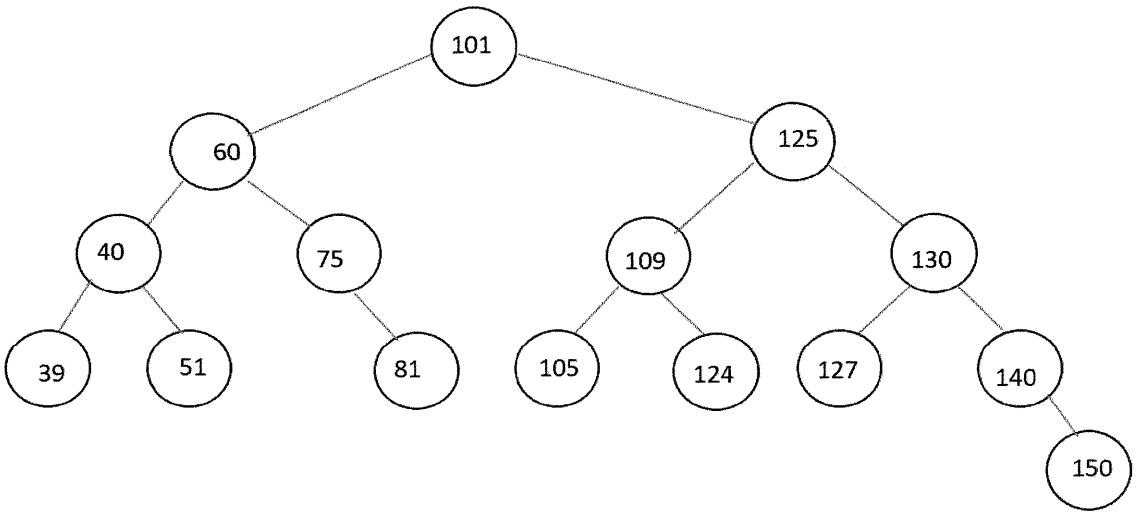
MARINHA DO BRASIL

DIRETORIA DE ENSINO DA MARINHA DO BRASIL

CP-CEM/ 2020 ENGENHARIA DE SISTEMAS DE COMPUTAÇÂO

1ª QUESTÃO (8 pontos)

a) (4 pontos)



b) (0,8 pontos)

4

c) (0,8 pontos)

3

d) (0,8 pontos)

4

e) (1,6 pontos)

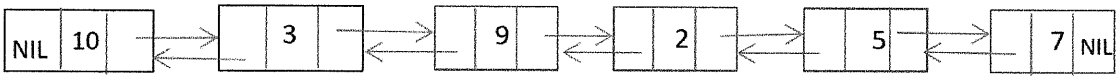
Valor	filho_esquerdo	Filho_direito

ou

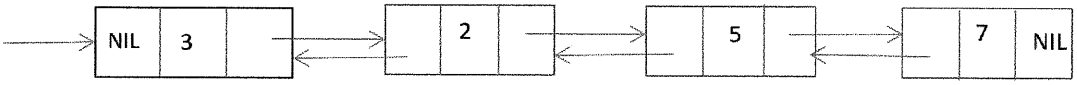
z.valorz.filho_esquerdoz.filho_direito

2ª QUESTÃO (8 pontos)

a) (4 pontos)



b) (2 pontos)



c) (2 pontos)

Θ (n)

3ª QUESTÃO (8 pontos)

a) (4 pontos)
A= 5, 9,2,7,6
p= 0
r= 4

j	A
0	5,9,2,7,6
1	5,9,2,7,6
2	5,2,9,7,6
3	5,2,9,7,6

A após execução da chamada de PARTICIONE:
A= 5,2,6,7,9

b) (2,4 pontos)
A=5,2,6,7,9
p= 0
r= 1

j	A
0	5,2,6,7,9

A após execução da chamada PARTICIONE:
A=2,5,6,7,9

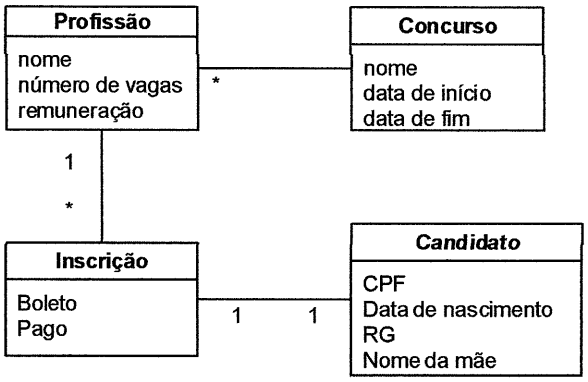
c) (1,6 pontos)
A=2,5,6,7,9
p= 0
r= -1

j	A
	Não executa nenhuma iteração j

A após execução da chamada:
A=2,5,6,7,9

4ª QUESTÃO (8 pontos)

a) (4 pontos)



b) (4 pontos)

```
CREATE TABLE concurso (
idConcurso INT NOT NULL AUTO_INCREMENT,
nome VARCHAR(100) NULL,
  inicio DATETIME NULL,
fim DATETIME NULL,
  PRIMARY KEY (idConcurso));


CREATE TABLE profissao (
idProfissao INT NOT NULL AUTO_INCREMENT,
idConcurso INT NOT NULL,
nomeVARCHAR(100) NOT NULL,
vagas INT NOT NULL,
remuneracaoDECIMAL(10,2) NOT NULL,
  PRIMARY KEY (idProfissao, idConcurso),
  CONSTRAINT concurso_profissao
    FOREIGN KEY (idConcurso)
    REFERENCES concurso (idConcurso));
```

5ª QUESTÃO (8 pontos)

a) (1,5 pontos)

Modelo incremental (Sommerville)

Modelo iterativo / evolucionário (Pressman)

Modelo iterativo incremental

b) (1,5 pontos)

As atividades são as mesmas. O que varia é a organização delas. As atividades são:

- Comunicação, Planejamento, Modelagem, Construção e Implantação (Pressman);
- Especificação, Projeto e Implementação (Desenvolvimento), Validação e Evolução (Sommerville); e
  Especificação de requisitos, Análise e Projeto, Implementação, Teste e Implantação.

c) (5 pontos)

```
CREATE TABLE hotel (
idHotel INT NOT NULL AUTO_INCREMENT,
nomeVARCHAR(100) NOT NULL,
telefoneVARCHAR(100) NOT NULL,
logradouroVARCHAR(100) NOT NULL,
numero INT NOT NULL,
cidadeVARCHAR(100) NOT NULL,
estadoCHAR(2) NOT NULL,
  cep CHAR(8) NOT NULL,
  PRIMARY KEY (idHotel));


CREATE TABLE suite (
idHotel INT NOT NULL,
numero INT NOT NULL,
precoDECIMAL(10,2) NOT NULL,
general INT NULL,
  PRIMARY KEY (numero, idHotel),
  CONSTRAINT hotel_suite
    FOREIGN KEY (idHotel)
    REFERENCES hotel (idHotel));

-- Outras opções:
-- - Duas tabelas para suíte.
-- - Uma tabela separada para endereço.
-- - General ser um binary.
-- - Apenas os ids precisam ser notnulls
```

6ª QUESTÃO (8 pontos)

a) (3 pontos)

Há diversas maneiras de implementar esse tipo de proteção, sendo que a maioria da atualidade passa por **assinatura do *firmware***[2 pontos]. O dispositivo deve ser carregado de fábrica com um certificado ou chave, que

utilizará para verificar a autenticidade do *firmware* no momento da atualização. Caso a assinatura não seja a esperada, o dispositivo rejeitará a atualização, impedindo o ataque. A implementação pode variar desde a utilização de módulos específicos em hardware (e.g. TPM, do inglês *Trusted Platform Module*) até a assinatura de programas individuais ou de mensagens (MAC, do inglês *MessageAuthenticationCode*). As desvantagens mais comuns são relativas ao custo [qualquer um dos três: 1 ponto]: **monetário** (espaço em memória, possíveis módulos de hardware para armazenar ou acelerar dados relativos a criptografia), **velocidade** (há o passo de verificação da assinatura que exige o cálculo de um *hash* e a verificação em si) e **energia** dispendida (devido ao custo computacional para realizar a verificação). Como a verificação só ocorre durante a atualização, o funcionamento normal do dispositivo não é afetado.

b) (2 pontos)
Para responder essa questão, é necessário conhecer o serviço de DNS, que utiliza primariamente o protocolo **UDP** [1 ponto (justificativa)]. O serviço é uma aplicação (camada 7), e pode usar TCP em uma segunda comunicação. Porém, a característica do ataque de utilizar todos os recursos disponíveis pula as camadas superiores, montando o pacote UDP diretamente e ignorando qualquer comunicação subsequente, portanto o alvo do ataque é o protocolo UDP. O protocolo UDP está na camada de **transporte** [1 ponto (resposta)] (camada 4).

c) (1 ponto)

Sim [0.5 ponto (resposta)], o provedor teve dificuldades (de fato, a reação do provedor levou cerca de 2h na primeira onda de ataques). É possível detectar o aumento repentino no volume de requisições, porém não é possível distinguir uma requisição DNS legítima de uma do atacante, pois **o ataque é proveniente de múltiplos IPs**[0.5 ponto (justificativa)] espalhados pela internet, de dispositivos que estão em redes legítimas. De fato, os dispositivos que participaram do ataque possivelmente fizeram ou poderiam fazer requisições DNS legítimas fora do período de ataque. Esse tipo de ataque é chamado de Ataque de Negação de Serviço Distribuído, ou DDoS (do inglês, *DistributedDenialof Service*).

d) (2 pontos)
É impossível bloquear o ataque apenas filtrando o acesso por IP (veja resposta ao item anterior). Também não é aceitável bloquear completamente o serviço, pois isso afetaria todos os clientes. Há duas alternativas bem conhecidas para mitigar o problema durante o ataque: (i) *blackhole* e (ii) filtro ativo. Na (i), todo o tráfego para os domínios afetados é redirecionado para uma rota nula, portanto todo o tráfego será descartado. Na (ii), é necessário abrir o pacote e verificar se a solicitação é para um subdomínio legítimo ou aleatório, sendo que nesse

último caso o pacote é descartado. Ambos os métodos são efetivos e previnem que outros domínios que não estão sob ataque tornem-se inacessíveis. O método (i) torna os domínios sob ataque completamente inacessíveis durante toda a sua vigência, enquanto o método (ii) descarta somente as requisições atacantes, permitindo que requisições legítimas ocorram normalmente dependendo da dimensão do ataque. Porém, o método (ii) requer recursos computacionais significativamente maiores que o método (i). Ambos os métodos podem ser aliados com bloqueio por IP (temporário ou não) ou delegação de filtragem (e.g. scrubbing), e precisam ser aplicáveis nos equipamentos de rede pertinentes (nesse tipo de ataque, normalmente no roteador de borda). Técnicas incomuns ou elaboradas (e.g. mapeamento de IPs por geolocalização, *caching*, etc.) podem ser aceitas como resposta, desde que justificadas corretamente. Qualquer uma das opções [1 ponto (explicar a solução)] e [1 ponto (explicar ao menos uma vantagem ou uma desvantagem)]

7ª QUESTÃO (8 pontos)

a) (4 pontos)

As arquiteturas apresentadas têm diferenças consideráveis entre si. O Intel é uma máquina CISC e o ARM uma máquina RISC, mas isto impacta pouco na diferença de desempenho. Ao observar a característica do algoritmo, vemos que é 95% matricial, portanto a presença de unidades de cálculo **vetorial** [4 pontos] com **capacidade de processamento matricial** [alternativa] pode resultar na diferença de tempo apresentada (o processador da fragata não tem ou não consegue utilizar esse recurso). Como não se conhece mais nada sobre os sistemas, a diferença arquitetural em relação ao cálculo matricial se torna a única opção viável.

b) (4 pontos)

Mesmo considerando o desconhecimento da origem do problema, **não é possível atingir 1s aumentando-se o número de cores** [1 ponto]. Segundo a lei de Amdahl, o desempenho paralelo do algoritmo aumentará no máximo 1/(1-0.95)=1/0.05=20 vezes. Se o desempenho teórico máximo é 20x, o algoritmo executará em 2s com um número infinito de cores [1 ponto para a justificativa]. Para 64 cores, o desempenho segundo a mesma lei será 1/(1-0.95+0.95/64)~**15.42** vezes [2 pontos].

8ª QUESTÃO (8 pontos)

a) (4 pontos)

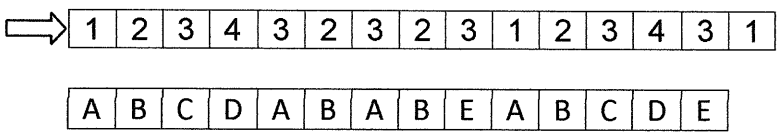
A diferença primordial de um SO para um RTOS está no escalonador. No RTOS, o escalonador **garante que a tarefa executará em um determinado período** e com uma **determinada latência** sempre que houver recursos disponíveis [3 pontos]. Para um algoritmo balístico com restrição de tempo de execução, essa característica não só **é recomendada** [1 ponto] como fortemente desejável. Há outros motivos favoráveis, mas esse é essencial em uma resposta correta.

b) (4 pontos)

Em relação ao sistema de arquivos, a troca também **é recomendada** [1 ponto] por três motivos, todos ligados a características de memórias tipo *flash* ou eeprom. O primeiro é que o acesso ao FS será mais **rápido** [0.5 ponto], pois o ext4 é otimizado para as buscas em discos mecânicos, inexistentes em memórias *flash*, que contam com **acesso direto aleatório** [0.5 ponto]. O segundo é que as memórias desse tipo precisam que um **bloco inteiro seja apagado** [0.5 ponto] antes de escrever algo, ação que toma um tempo considerável; os FSs convencionais escrevem frequentemente no mesmo local da memória, enquanto os preparados para flash usam outro bloco livre e apagam os blocos somente quando necessário, diminuindo o tempo médio de escrita [0.5 ponto]. O terceiro motivo também é baseado na escrita; como há uma quantidade limitada de escritas possíveis em um bloco de memória *flash*, a escrita feita por FSs preparados para esse tipo de memória **balanceia a escrita** [0.5 ponto]entre os blocos, de forma que a memória não seja comprometida porque um único bloco atingiu o limite de apagamentos-escritas, aumentando assim sua **vida útil** [0.5 ponto].

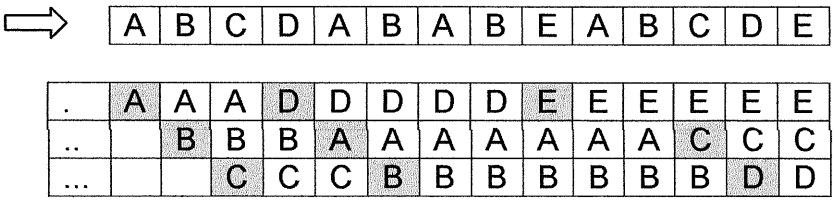
9ª QUESTÃO (8 pontos)

a) (1,5 pontos)



O término do programa nessas condições pode ser considerado normal, porque ocorre em uma ocasião em que o programa se encontra no estado 1, que é o único estado considerado como sendo ponto de término normal do programa, de acordo com a especificação fornecida.

b) (2,5 pontos)



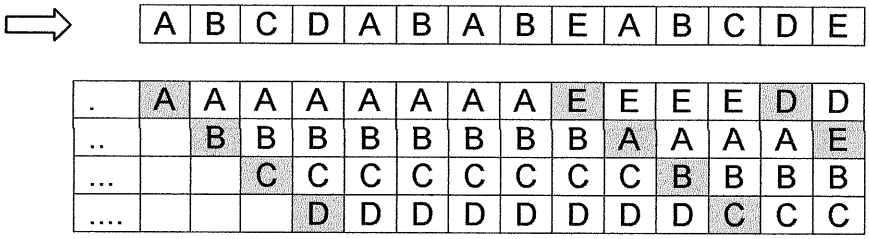
⇨	A	B	C	D	A	B	A	B	E	A	B	C	D	E
.	A	A	A	D	D	D	D	D	E	E	E	E	E	E
..		B	B	B	A	A	A	A	A	A	A	C	C	C
...			C	C	C	B	B	B	B	B	B	B	D	D

A linha do cabeçalho é apenas a transcrição do resultado da questão anterior.

As três primeiras interrupções, causadas pelas três primeiras referências (páginas A, B e C) apenas copiam as páginas A,B,C do disco para a memória (destacadas com fundo mais escuro).
As três referências seguintes (páginas D, A e B) também causam interrupções que desalojam A, B e C, respectivamente, visto que essas referências acontecem em ocasiões em que as páginas referenciadas não se encontram presentes na memória. Já as duas seguintes (A e B) encontram essas páginas na memória e não causam interrupções. Em seguida, uma referência à página E causa uma interrupção, e a leitura de E é feita para o bloco que continha a página mais antiga, ou seja, D, que é substituída por E. Prosseguindo, A e B não causam interrupções e, em seguida, C e D desalojam A e B, respectivamente, terminando o programa com uma referência à E, que se encontra na memória e não causa interrupção.
Assim, essas seis últimas interrupções devem:

- no pior caso - gravar no disco cópia da página substituída, além de trazer para a memória física cópia da página recém-referenciada, e
- no melhor caso - caso a página presente na memória não tenha sofrido alterações, não haverá necessidade de gravá-la no disco. Assim, no melhor caso, temos 3+6=9 movimentos de 4096 bytes (apenas do disco para a memória), enquanto, no pior caso, serão 3+2x6=15 movimentos de blocos, (sendo 9 do disco para a memória e mais 6 da memória para o disco).

c) (1,5 pontos)



⇨	A	B	C	D	A	B	A	B	E	A	B	C	D	E
.	A	A	A	A	A	A	A	A	E	E	E	E	D	D
..		B	B	B	B	B	B	B	B	A	A	A	A	E
...			C	C	C	C	C	C	C	C	B	B	B	B
....				D	D	D	D	D	D	D	D	C	C	C

d) (2,5 pontos)

Tratando-se do mesmo programa, o número de interrupções por falta de página (10 no segundo caso, contra 09 no primeiro) cresceu do caso anterior para este, confirmando a manifestação da anomalia: aumentou-se a quantidade de memória disponível, mas isso não só não reduziu, como aumentou o número de interrupções de referências a páginas faltantes.

Para evitar os efeitos da anomalia, é preciso alterar a política de substituição de páginas: para evitar a remoção de páginas antigas na memória, porém muito referenciadas pelo programa, dá-se preferência a algoritmos inspirados no uso de pilhas, como por exemplo, o LRU (*leastrecentlyused*: dentre as páginas presentes na memória física, opta por substituir em primeiro lugar aquelas páginas que há mais tempo tenham permanecido ociosas na memória, ou seja, aquelas que há mais tempo não tenham sido referenciadas).

10ª QUESTÃO (8 pontos)

a) (2,0 pontos)

```
P→ D E $
D →var N S; | ε
E → A | B | ε
A →N := N
B →{ E L }
L →; E L | ε
S →, N S | ε
N →x | y
```

b) (1,0 pontos)

não-terminais:

{P D E A B L S N}

(porque são os símbolos que são substituídos, ao longo da análise, pela ação das regras gramaticais)

terminais:

{var ; :={},xy$}

(porque são os elementos a serem encontrados na cadeia de entrada no momento da análise)

raiz da gramática:

`P`

(porque é o conteúdo inicial da pilha de análise descendente)

c) (2,0 pontos)

Regular: não é.

Porque o não-terminal `B` provoca aninhamentosauto-recursivos, tais como:

`B ⇒{ E L }⇒{ B L }`

d) (3,0 pontos)

posicionar um ponteiro ao início do texto-fonte;

loop: de acordo com o caractere de entrada apontado:

símbolos compostos: retornar os tokens
`var` (uma palavra reservada) ou
`:=` (token de atribuição);
avançar;
go to loop;

símbolos isolados: retornar o respectivo token: `; { } , x y`
avançar,
go to loop;

fim de texto `$` retornar o token correspondente;
terminar a análise léxica;

qualquer outro
nada retornar;
avançar;
goto loop;