1. Árvores de decisão. Nesta questão, vamos explorar o conceito de árvores de decisão.

(a)  
LDA 20 #00 0001 00010100  
SUB 5 #01 0100 00000101  
JP 05 #02 0110 00000101  
LDA 8 #03 0001 00001000  
JMP 11 #04 1000 00001011  
LDA 21 #05 0001 00010101  
SUB 3 #06 0100 00000011  
JP 10 #07 0110 00001010  
LDA 7 #08 0001 00000111  
JMP 11 #09 1000 00001011  
LDA 6 #10 0001 00000110  
STR 30 #11 0010 00011110  
HLT #12 0000

(b)

2. Construção de atributos (features). Você está desenvolvendo um sistema embarcado de classificação de sons para um microcontrolador de baixo custo, como parte de um projeto de IA embarcada. O sistema não possui unidade de ponto flutuante, e os dados sensoriais são recebidos como inteiros entre os endereços de memória 20 a 22. Cada posição representa a intensidade do som em diferentes janelas de tempo.

(a)  
LDA 20 #00 0001 00010100  
STR 40 #01 0010 00101000  
STR 41 #02 0010 00101001  
ADD 21 #03 0011 00010101  
ADD 22 #04 0011 00010110  
STR 30 #05 0010 00011110  
LDA 21 #06 0001 00010101  
SUB 40 #07 0100 00101000  
JP 11 #08 0110 00001011  
LDA 21 #09 0001 00010101  
STR 40 #10 0010 00101000  
LDA 22 #11 0001 00010110  
SUB 40 #12 0100 00101000  
JP 16 #13 0110 00010000  
LDA 22 #14 0001 00010110  
STR 40 #15 0010 00101000  
LDA 21 #16 0001 00010101  
SUB 41 #17 0100 00101001  
JN 21 #18 0111 00010101  
LDA 21 #19 0001 00010101  
STR 41 #20 0010 00101001  
LDA 22 #21 0001 00010110  
SUB 41 #22 0100 00101001  
JN 26 #23 0111 00011010  
LDA 22 #24 0001 00010110  
STR 41 #25 0010 00101001  
LDA 40 #26 0001 00101000  
SUB 41 #27 0100 00101001  
STR 31 #28 0010 00011111  
ABS 31 #29 1010 00011111  
STR 31 #30 0010 00011111  
HLT #31 0000

(b) É vantajoso já que a transferência de dados entre dispositivos é menor, e os valores iniciais não são expostos como o resultado final. Devido a menor transferência de dados, o gasto de energia e banda de internet tende a ser menor.

(c) Com os valores separados na memória o programa não pode carregar da memória RAM vários valores em sequência. Por exemplo, tendo uma lista de 4 bytes, o programa pode copiar todos os valores com uma única chamada movl para um registro de 4 bytes e ler cada bit individualmente, salvando assim em acessos à memória RAM. Localidade temporal é quando se pode reutilizar valores já armazenados em registros ou no cache, e localidade espacial é quando se pode ler vários valores que serão usados em uma única operação.

3. Nessa questão, vamos estudar como a hierarquia de memória impacta o desempenho desses sistemas.

(a) Os registros são espaços de memória para uso direto do processador, sendo a memória de acesso mais rápido porém de menor capacidade.  
O cache é gerido automaticamente pelo hardware e está em segundo lugar em velocidade, sendo usado para armazenar valores utilizados com frequência. O cache está distribuído em níveis L1, L2 e L3, sendo o L1 o mais rápido mas de menor capacidade, com a velocidade diminuindo e a capacidade aumentando ao decorrer dos níveis. O cache possui mais espaço do que os registros mas não passa de alguns MiB.  
A memória RAM serve para armazenar os aplicativos em execução, sendo gerenciada pelo S.O.. A memória RAM possui acesso mais lento do que o cache mas possui mais capacidade, comumente alguns GiB.  
O disco rígido guarda os dados para longo prazo, sendo dos tipos de memória mencionados, o único que não depende de energia externa para manter seu estado. Possui a maior capacidade devido ao armazenamento em discos ou em células de memória flash. Porém o disco rígido tem o acesso mais lento devido também ao seu método de armazenamento de dados.

(b-a) Porque o arquivo pode já estar carregado na memória RAM, não precisando ser lido do disco.

(b-b) É reutilizar dados já carregados previamente devido a um uso passado.

(b-c) É carregar vários elementos a serem usados de uma só vez, como carregar dois i32 com um movq.

(c) Desfragmentar a memória, armazenando dados de forma continua facilitando “cache hits”.

4. Pipelines em LLMs. Grandes Modelos de Linguagem (LLMs), como o ChatGPT, são treinados processando bilhões de exemplos e realizando trilhões de operações matemáticas.

(a) {Tempo de execução das 5 etapas em uma entrada} \* 12

(b) 12 + 3 \* {Tempo de execução das 5 etapas em uma entrada}