Árbol B de orden m = 5:

* 1. a. ¿Cuál es el número mínimo y máximo de claves que puede tener un nodo interno (no raíz)?
* Cada nodo puede tener máximo m – 1 claves.
* Cada nodo (excepto la raíz y hojas) debe tener entre (m/2) y m hijos, es decir entre [(m/2) – 1] (mínimo) y [m – 1] (máximo) claves.
* En este caso: m = 5 ⇒ (5 / 2) ≈ 3
* Mínimo de claves en nodo interno: [(m/2) – 1] ⇒ 3 - 1 = 2
* Máximo de claves en nodo interno: [m – 1] ⇒ 5 – 1 = 4
  1. b. ¿Cuál es el número mínimo y máximo de hijos que puede tener un nodo interno?
* Máximo: m hijos ⇒ 5 hijos.
* Mínimo (excepto raíz): [m/2] hijos ⇒ 3 hijos.
  1. c. ¿Cuál es el número mínimo de claves que puede tener la raíz (si no es hoja)?
* Si la raíz no es hoja, debe tener al menos 2 hijos ⇒ implica al menos 1 clave
  1. d. Justifique por qué los árboles B son especialmente adecuados para manejar grandes volúmenes de datos almacenados en disco.
* Alta fan-out (muchos hijos por nodo): Cada nodo almacena muchas claves, lo que disminuye la altura del árbol.
* Pocos accesos a disco: Si la raíz se mantiene en RAM, una búsqueda necesita 1 o 2 accesos a disco, en lugar de 20 o más como en AVL o ABB.
* Balanceo automático: Todos los caminos de la raíz a las hojas tienen la misma longitud, garantizando eficiencia uniforme.
* Uso eficiente del almacenamiento: Se diseñan para que cada nodo coincida con el tamaño de un bloque de disco (por ejemplo, 1 KB o 4 KB).
* Los árboles B son ideales para bases de datos y sistemas de archivos porque minimizan la cantidad de accesos a disco, maximizan el uso de cada bloque leído y mantienen el árbol balanceado eficientemente.