

Motivación y Coherencia (5)	
1	Menciona que la motivación . Resume que es aprender a implementar optimizaciones usando hashing y a diseñar estrategias de backtracking para problemas de restricciones.
1	Indica la relevancia que tiene hashing . Ideal con un ejemplo de otro problema.
1	Indica la relevancia que tiene backtracking . Ideal con un ejemplo de otro problema.
1	Conclusión coherente . Cierre y resumen claro de lo mencionado anteriormente.
1	Coherencia general . El informe es coherente, empleando un razonamiento profesional, y atinente a los contenidos del curso.
Rolling DNA Cores (5 puntos máximo) Si utilizan una técnica distinta de hash, se corrige sin esta pauta.	
2 <small>se espera 2 puntos</small>	<p>Explican claramente la complejidad, y se entiende por qué no es $O(N M Q)$</p> <p>Llegan correctamente a $O(N Q)$ explicando el porqué (idea de “sliding window”) Queda claro porque no es necesario comparar string por string. 0 puntos si entregan la complejidad, pero no la explican.</p> <p>https://courses.csail.mit.edu/6.006/spring11/rec/rec06.pdf</p>
1 <small>se espera 0.5 puntos (se da todo de estos puntos en el anexo 10))</small>	<p>Explican mejorar complejidad, como $O(N + MQ)$ o $O(N + MQ + K + (\text{Factor Colisiones}))$. Se entiende como usan una EDD, como una tabla o árboles, para reducir el tiempo. No cuenta como mejora aumentar la tabla de hash.</p> <p>https://en.wikipedia.org/wiki/Rabin%E2%80%93Karp_algorithm#Multiple_pattern_search</p>
1 <small>se espera 1 puntos</small>	<p>Explica como es la función de hash.</p> <p>Indica que la función de hash se divide en 2: de 0 a M (sin roll) y de M a N (con roll).</p> <p>Pueden entregar la ecuación por casos o indicando que se aplica una función de hash distinta para los substring y el pedazo inicial, y otra para seguir la iteración sobre el string.</p>
1 <small>se espera 1 punto</small>	<p>Explican el dominio de la función de hash. Deben indicar que se mapea la secuencia de caracteres a números a número de entrada y salida de [0, 4] quedando en base 4. Puede ser similar (por ejemplo, [1, 4]).</p> <p>0.5 si menciona un mapeo de char a ints, pero entrega otros números sin explicación.</p>
2 <small>se espera 1 punto</small>	<p>Indican que se pueden almacenar o trabajaren memoria al menos 2 de lo siguiente. Dar puntaje parcial.</p> <ul style="list-style-type: none"> • Guardar los valores previamente calculados (no calcular rolling hash Q veces) <ul style="list-style-type: none"> ◦ Esto puede ser guardando los valores en un arreglo, tabla de hash, etc • Como evita colisiones si es que aplica <ul style="list-style-type: none"> ◦ Lista ligada que referencia a las slides para comparar • Consideración general al elegir módulo o el tamaño de la tabla de hash <ul style="list-style-type: none"> ◦ No cuenta indicar un número primo sin mencionar distribución o memoria ◦ Idealmente, mencionar equilibrio entre colisiones y uso de memoria • [0.5] guardar el valor $B^{(L-1)}$ para calcularlo inicialmente en vez de N^2 veces.
1	Indica que no hay colisiones cuando $\log_2(\text{Tamaño Base}) \times \text{Substring} \leq \text{Modulo Hash}$

Rompecabezas Cósmico de Entregas (5 puntos máximo)	
1	Explica la iteración en el espacio. Deberá ser entre fila $\in [-2, \text{FILAS}]$, col $\in [-2, \text{COLUMNAS}]$ o similar , para poder ver toda posición que puede tomar un paquete de 3 x 3 dentro de un tablero de FILAS x COLUMNAS. No entregar puntaje si no se profundiza más que “iterar por toda la bodega o tablero”.
2	Indica cuál es el paso recursivo , explicando cuál es la estrategia para retroceder . Debería ser similar a “poner, probar y sacar”, explicando bien los 3 pasos. 1 punto máximo si no se menciona como se recuerda que es lo que se debe sacar. Esto puede ser, por ejemplo, con el índice del paquete actual, que queda guardado en el scope de la función.
1	Indica cuál es el caso de término . Deberá ser similar a “cuando no hay más paquetes”.
1	Se explica las optimizaciones de backtracking 1: Podas y Propagación Estas optimizaciones suelen quedar implícitas en este problema. Podas es no poner un paquete sobre otro , y propagación es guardar los espacios que estén ocupados . Explica estas dos optimizaciones. No es necesario mencionarlas explícitamente.
1	Indica explícitamente que son las podas y propagación en este problema.
1	Se explica las optimizaciones de backtracking 2: Heurísticas Se explica como se podría optimizar backtracking dado una estrategia que permita encontrar más rápidamente la solución. Esto puede ser, por ejemplo, una iteración más inteligente sobre el tablero u ordenar qué paquetes añadir.
Patrones de duplicación de Bender (5 puntos máximo)	
Si utilizan una técnica distinta de hash, se corrige sin esta pauta.	
1	Explica y muestra la función de hash , indicando cuáles son las entradas de la función. 0.5 si no menciona la transformación del dominio (caracteres) a números para la función.
2	Explica como se optimiza la búsqueda, usando hash. Se espera que se explique: <ul style="list-style-type: none"> • Como se recorre el árbol para emplear la función de hash. • La complejidad (no en detalle) de aplicar hash, considerando lo anterior • Como se guarda cada subárbol en la tabla de hash. 1 punto si ni está claro los puntos anteriores. 0 si no solo se menciona por encima la iteración. Por ejemplo, decir que se usó iteración o recursión, pero sin mencionar como aplica hash y se guardan los resultados.
1	Explica como se garantiza el orden de los resultados. Esto puede ser garantizado con una lista ligada ordenada por cada árbol idéntico u ordenando los resultados.
1	Explica qué se hace cuando hay colisiones en la tabla de hash (función comparar árboles).
1	Replica que se hace con árboles idénticos . Se espera que indique que no es necesario almacenar dos árboles idénticos, evitando comparaciones adicionales.
1	Explica como se puede usar la altura o número de nodos del subárbol para evitar colisiones .