

Pregunta 2 Presenta un algoritmo para encontrar un rey y analiza su complejidad y correctez.

Algoritmo El algoritmo es el siguiente: Recorrerse el torneo buscando el elemento que tiene exvecindad máxima.

```
def busca_rey(T)
    maximo = 0
    rey = None
    for {v in T.vertices}:
        if len(v.adjacent) > maximo:
            maximo = len(v.adjacent)
            rey = v
    return rey
```

Este algoritmo busca exhaustivamente sobre los vértices, por lo que necesariamente encuentra aquel de máxima exvecindad.

Asumiendo que la representación de G es la de adyacencia con listas ligadas, el algoritmo es del orden de $O(n + m)$, pues cada vértice se recorre una vez y ya en el y solo se toma el tamaño de la lista de adyacencia, lo cual es en el caso de una lista ligada $O(m)$.