2. Consideramos el siguiente escenario: tenemos un conjunto de n ciudades con distancias mínimas entre ellas (verifican la desigualdad triangular). Queremos seleccionar un subconjunto C de k ciudades en las que queremos ubicar un centro comercial. Asumiendo que las personas que viven en una ciudad compraran en el centro comercial más próximo se quiere buscar una ubicación C de manera que todas las ciudades tengan un centro comercial a distancia menor que r en la que intentamos minimizar r sin perder cobertura. Para ello diremos que C es un r-recubrimiento si todas las ciudades están a distancia como mucho r de una ciudad en C. Sea $r(C)$ el mínimo r para el que C es un r-recubrimiento. Nuestro objetivo es encontrar C con k vértices para el que $r(C)$ es mínimo.

(a) Demuestra que si $k \geq n$, la solución formada por todas las ciudades es óptima.

A partir de ahora asumiremos que $k \leq n$.

(b) Suponiendo que S es el conjunto de ciudades, considera el siguiente algoritmo:

Seleccionar cualquier ciudad $s \in S$ y define $C = \{s\}$
while $|C| \neq k$ **do**
  seleccionar una ciudad $s \in S$ que maximiza la distancia de s a C;
  $C = C \cup \{s\}$
return C

Demuestra que es un algoritmo de aproximación polinómico con tasa de aproximación 2.

a) Si $k \geq n$, podemos hacer que nuestro subconjunto C sea todas las ciudades, por lo tanto esta sería una solución óptima ya que cada ciudad tiene una distancia $r=0$, ya que en el conjunto C están incluidas todas las ciudades.

b) Primero demostramos que el algoritmo se ejecuta en tiempo polinómico.

El bucle while se repite k veces, ya que $|C| \neq k$, entonces en cada bucle se hace una búsqueda de las n ciudades en que se maximiza la distancia de s a C.

Por lo tanto el coste $O(nk)$.

b) Para demostrar que el algoritmo devuelve una solución que es como máximo el doble de la óptima, consideramos que la solución óptima C^* de tamaño k. Para cualquier ciudad $s \in S$, sea $d(s, C^*)$ la distancia más corta de s a cualquier ciudad en C^*. Entonces

$d(s, C) \leq 2\, d(s, C^*)$, ya que s se selecciona como la ciudad que maximiza

La distancia a C, $d(s,C) \geq d(s,C^*)$. Por lo tanto cualquier ciudad en S estará a distancia como máximo $2r(C^*)$ de alguna ciudad en C. Por lo tanto es un algoritmo de aproximación 2.

$$d(s,C) \leq 2\,d(S,C^*)\;;$$

- Suponemo $C' = C \cup \{s\}$
- Sea x tq $d(x,C') > d(x,C^*)$

Sabemos que s es la ciudad que maximiza la distancia de s a C, $d(s,C)$.

Por lo tanto tenemos que $d(s,C) = d(s,C')$ y $d(s,C^*) \leq d(s,x)$

Como S verifica la desigualdad triangular:

$d(x,C') < d(x,s) + d(s,C')$ y $d(x,C^*) < d(x,s) + d(s,C^*)$

Sumando las desigualdades $d(x,C') + d(x,C^*) \leq 2d(x,s) + 2d(s,C^*) \leq 4d(s,C)$