

1. (+20pts) Se sabe que es posible demostrar que TSP es NPC usando el problema HAM-CYCLE. Para esto se muestra una reducción polinomial  $HAM - CYCLE \leq_p TSP$ , y demostrando que TSP pertenece a NP. De argumentos basados en la teoría vista en clase para justificar o negar la siguiente afirmación, explore consideraciones y puntos importantes para cualquier caso:

Lo anterior quiere decir que TSP también se puede reducir a HAM-CYCLE y en general cualquier problema NPC se puede reducir a cualquier problema NPC. Lo mismo se puede decir para los problemas NP.

2. (+16pts) Se sabe que si  $P \neq NP$  entonces para cualquier constante  $1 \leq p$ , no existe un algoritmo aproximado en tiempo polinomial con ratio de aproximación  $p$  para el problema generar de TSP.

- (+8pts) Muestre como en tiempo polinomial se puede traducir una instancia del problema de TSP en una instancia donde la función de costo satisface la desigualdad triangular. Las dos instancias deben tener el mismo conjunto de tours óptimos.
- (+8pts) Asuma que  $P \neq NP$  explique como esto no contradice el teorema mencionado.

3. (+14pts) Proponga un algoritmo (*Java, Python, C, ...*) basado en alguna heurística vista en clase que solucione el siguiente problema, la entrada del algoritmo debe ser  $n:int$  y debe mostrar por consola la respuesta adecuada:

El conjunto  $\{1, 2, 3, \dots, n\}$  contiene un total de  $n!$  permutaciones distintas. Al enlistar y etiquetar todas las permutaciones en orden lexicográfico ascendente, obtenemos la siguiente secuencia para  $n = 3$ :

123, 132, 213, 231, 312, 321

Proporcione un algoritmo (+10pts) eficiente que devuelva la  $k$ -ésima de las  $n!$  permutaciones en esta secuencia, dados los valores de  $n$  y  $k$ . Para ser eficiente, no debería construir las primeras  $k - 1$  permutaciones en el proceso. (+4pts) Cual es la complejidad de la solución.