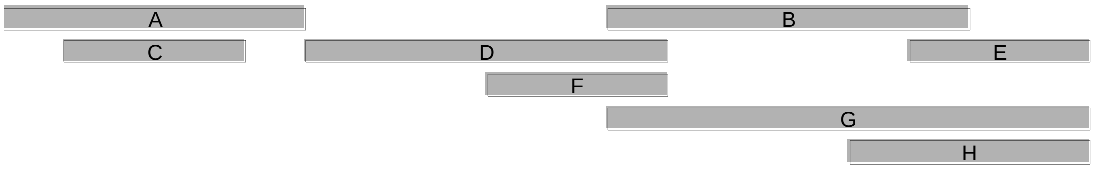
**ADA – Tarea 3**
**Xavier Garzón**

***Greedy Schedule***

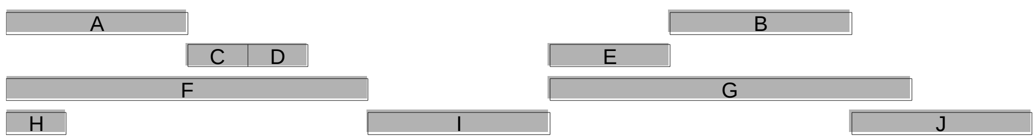
***a)***




El algoritmo nos dice cuál intervalo debemos elegir pero no nos dice nada si dos o más intervalos cumplen la condición de terminar de último, por tanto pueden dar resultados distintos.

- ***E-F-A***
- ***G-A***

***b)***




El algoritmo nos dice cuál intervalo debemos elegir pero no nos dice nada si dos o más intervalos cumplen la condición de terminar de iniciar de primero, por tanto pueden dar resultados distintos.

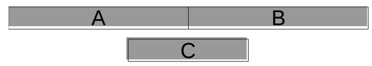
- ***A-C-D-I-E-B***
- ***F-I-E-B***

***c)***

Sea $B \subseteq A$ un agrupamiento óptimo de cursos donde A es una colección de cursos (finita) tal que $a \in A$ y $a$ es un curso con tiempo de inicio último entre todas los cursos de $A$.

- Si $a \in B$ entonces $a$ hace parte de una solución óptima.
- Si $a \notin B$ entonces sea $b \in B$ un curso con tiempo de inicio último en $B$. Se sabe que $b \neq a$. Sea $\mathbf{B'} = (\mathbf{B} \mid \{\mathbf{b}\}) \cup \{\mathbf{a}\}$, note que $|\mathbf{B'}| = |\mathbf{B}|$ y que incluir $a$ en $B$ no causa conflictos porque $a$ es el curso de inicio último en $A$

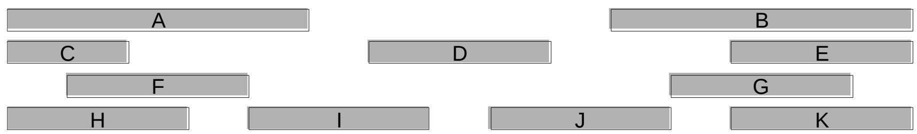
***d)***




Usando el algoritmo obtenemos una respuesta con un curso, pero es posible tomar dos cursos.

- ***C***
- ***A-B***

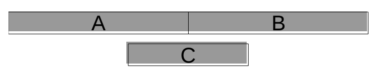
**e)**




Usando el algoritmo obtenemos una respuesta con tres cursos, pero es posible tomar cuatro cursos.

- ***F-D-G***
- ***H-I-J-K***

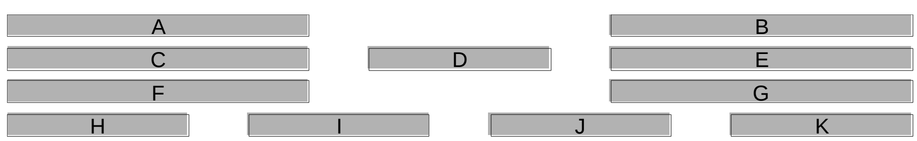
***f)***




Usando el algoritmo obtenemos una respuesta con un curso, pero es posible tomar dos cursos.

- ***C***
- ***A-B***

***g)***




Usando el algoritmo obtenemos una respuesta con tres cursos, pero es posible tomar cuatro cursos.

- ***F-D-G***
- ***H-I-J-K***

***i)***

Sea $B \subseteq A$ un agrupamiento óptimo de cursos donde $A$ es una colección de cursos (finita) tal que $a \in A$ y $b \in B$.

- Si $b \subseteq a$ entonces existe una solución óptima donde $a$ puede ser descartado y ser tomado $b$.
- Si $b$ no está contenido en $a$ entonces sea $b$ un curso que termina último. Tenemos que si $b$ termina último entonces $b$ inicia último por tanto no habría conflictos.

***Stabbing Points***

```python
def BinarySearch(A, lo, hi, x):
    """
    Búsqueda binaria que encuentra el primer número
    mayor a x
    """
    ans=-1
    if lo==hi:
        if A[lo][1]>x:
            ans=A[lo][1]
    elif len(A)!=0:
        while lo+1<hi:
            mid=(lo+hi)>>1
            if A[mid][1]<=x:
                lo=mid+1
            else:
                hi=mid
        if A[lo][1]>x:
            ans=lo
        elif A[hi][1]>x:
            ans=hi
    return ans

def solve(L,R,P):
    """
    Objetivo:
        Encontrar la cantidad de mínima de cortes que se pueden hacer
        con todos los elemento de P respecto a los rangos de L y R
    Entrada:
        L: arreglo con los puntos de inicio de cada segmento
        R: arreglo con los puntos de fin de cada segmento
        P: arreglo con los puntos que se espera crucen los segmentos
            creados con L y R posición a posición
    Salida:
        ans: arreglo de tuplas donde el primer valor de las tuplas es
                un elemento de P y el segundo valor es la cantidad de
                segmentos que el elemento de P cortó
    """
    ans=[]
    arr=zip(L,R);arr.sort(key=lambda item: (item[1]))
    for i in P:
        j=BinarySearch(arr,0,len(R),i)
        cnt=0
        if j!=-1:
            while (not (arr[j][0]<=i<=arr[J][1])) and j<len(R):j+=1
            while (arr[j][0]<=i<=arr[J][1]) and j<len(R):
                j+=1;cnt+=1
            if cnt!=0:ans.append((i,cnt))
    return ans
```