

- Se desea generar todas las permutaciones de n enteros distintos.
- Se puede utilizar la siguiente solución:
 - Se hará búsqueda con retroceso a través de todas las cadenas n -arias de longitud n , podando aquellas que no sean permutaciones.
 - Para ello se mantendrá un arreglo $U[1, \dots, n]$ donde $U[i]$ es verdadero si y sólo si m no ha sido usada.
 - Se mantendrá a la permutación actual en un arreglo $A[1, \dots, n]$.
 - El objetivo es llamar a $procesa(A)$ exactamente una vez para cada permutación almacenada en $A[1, \dots, n]$.

- Observe que este algoritmo es muy parecido al que usamos para caminos hamiltonianos.
- Esto se debe a que un camino hamiltoniano es una permutación de los vértices de la gráfica.

Procedimiento $permuta(m)$

- Si $m = 0$ entonces $procesa(A)$ y si no:
- Para $j = 1$ hasta n haz:
 - Si $U[j]$ es verdadero entonces:
 - Haga $U[j]$ falso.
 - $A[m] = j$.
 - $permuta(m - 1)$.
 - Haga $U[j]$ verdadero.

- Son i entradas escogidas de entre n y permutadas de alguna forma.
- Entonces existen $\binom{n}{i} i!$ formas de llenar esa parte.
- Y por lo tanto la cantidad de veces que se hace la pregunta es

$$\begin{aligned} \sum_{i=0}^{n-1} \binom{n}{i} i! &= n \sum_{i=0}^{n-1} \frac{n!}{(n-i)!} \\ &= n \cdot n! \sum_{i=1}^n \frac{1}{i!} \\ &\leq (n+1)! \cdot (e-1) \\ &\leq 1.719(n+1)! \end{aligned}$$

- Y por lo tanto toma tiempo $O((n+1)!)$ (mucho mejor que $O(n^n)$).

- Sea $T(n)$ la cantidad de intercambios hechos por $permuta(n)$.
- Entonces $T(1) = 0$ y $T(n) = nT(n-1) + 2(n-1)$ para $n \geq 2$.
- Demostraremos por inducción en n que $T(n) \leq 2n! - 2$.
- Esto es claramente cierto para $n = 1$. Si lo suponemos cierto para n :

$$\begin{aligned} T(n+1) &= (n+1)T(n) + 2n \\ &\leq (n+1)(2n! - 2) + 2n \\ &= 2(n+1)! - 2. \end{aligned}$$

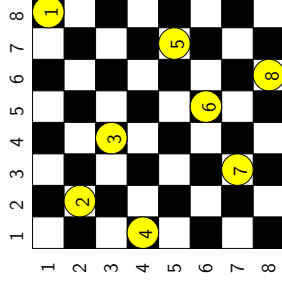
- Por lo tanto $T(n) \in O(n)$ y el algoritmo es óptimo.

- Este algoritmo claramente se ejecuta en tiempo $O(n^n)$.
- Pero este análisis no es el mejor posible porque no considera la poda.
- Calculemos la cantidad de veces que se pregunta si $U[j]$ es verdadero.
- ¿Cuál es la situación cuando se hace esa pregunta?
- Para alguna $0 \leq i < n$ el algoritmo:
 - ha llenado las últimas i entradas de A con parte de una permutación y
 - está intentando n candidatos para llenar la siguiente entrada de A .
- ¿Cómo se ven esas i entradas?

Procedimiento $permuta(m)$

- Si $m = 1$ entonces $procesa(A)$, si no:
 - $permuta(m-1)$.
 - Para $i = m-1$ hasta 1 haz:
 - Intercambia $A[m]$ con $A[i]$.
 - $permuta(m-1)$.
 - Intercambia $A[m]$ con $A[i]$.

- ¿De cuántas formas se pueden poner n reinas en un tablero de ajedrez de $n \times n$ de modo que ninguna reina ataque a ninguna otra?
- Si almacenamos en $A[i]$ el renglón en el que se encuentra la reina de la columna i obtenemos una permutación.



Procedimiento $hamilton(m)$

- Si $m = 0$ entonces $procesa(H)$ y si no:
- Para $j = m$ hasta 1 haz:
 - Si $M[H[m+1], H[j]]$ es verdadero entonces:
 - Intercambia $H[m]$ con $H[j]$.
 - $hamilton(m-1)$.
 - Intercambia $H[m]$ con $H[j]$.

Procedimiento $procesa(H)$

- Si $M[H[1], n]$ es verdadero imprime(H).