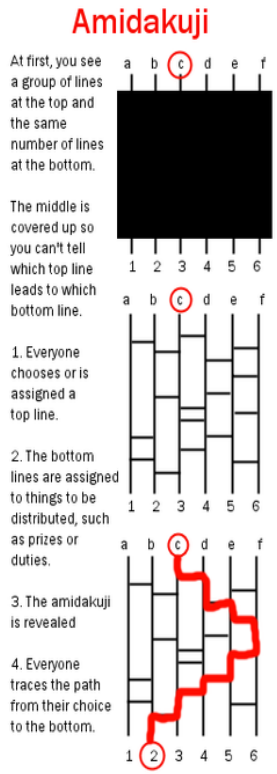
Amidakuji, sorting algorithms, & permutations:

From recreational mathematics to research mathematics

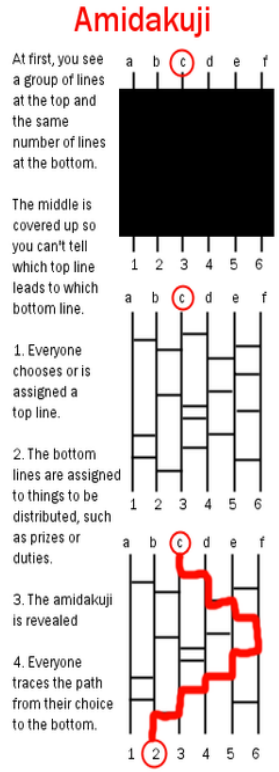
Chi-Kwong Li (ckli@math.wm.edu)
Department of Mathematics
The College of William and Mary

Amidakuji/Ghost Leg Drawing



Amidakuji/Ghost Leg Drawing

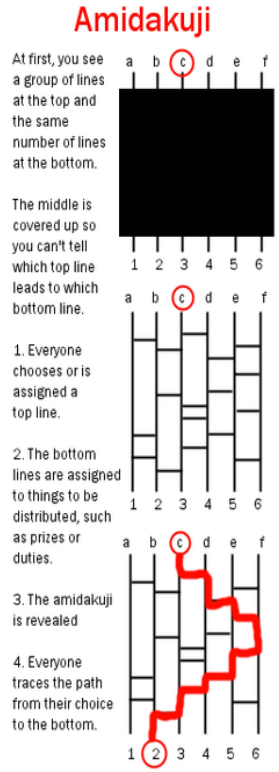
It is a scheme for assigning n people $P_1, \ldots, P_n$ to n jobs $J_1, \ldots, J_n$ "randomly".



Amidakuji/Ghost Leg Drawing

It is a scheme for assigning n people $P_1, \ldots, P_n$ to n jobs $J_1, \ldots, J_n$ "randomly".

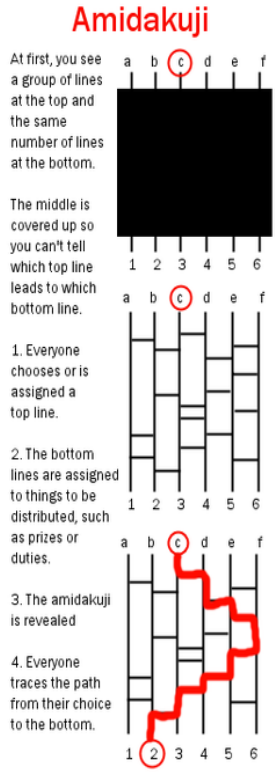
- Draw vertical lines from P_i to J_i from $i = 1, \ldots, n$.



Amidakuji/Ghost Leg Drawing

It is a scheme for assigning n people $P_1, \ldots, P_n$ to n jobs $J_1, \ldots, J_n$ "randomly".

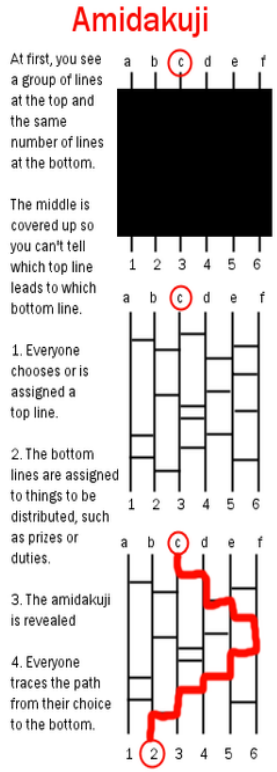
- Draw vertical lines from P_i to J_i from $i = 1, \ldots, n$.
- Draw some horizontal line segments randomly between any two vertical lines that are next to each other so that no horizontal lines meet.



Amidakuji/Ghost Leg Drawing

It is a scheme for assigning n people $P_1, \ldots, P_n$ to n jobs $J_1, \ldots, J_n$ "randomly".

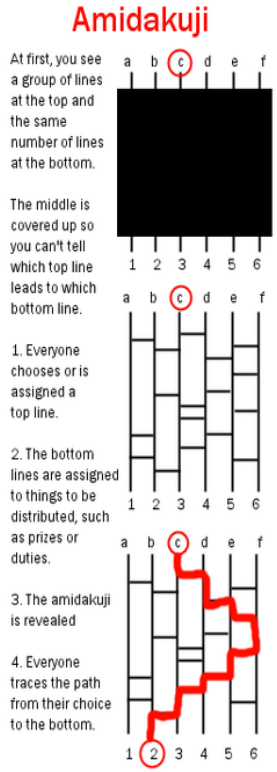
- Draw vertical lines from P_i to J_i from $i = 1, \ldots, n$.
- Draw some horizontal line segments randomly between any two vertical lines that are next to each other so that no horizontal lines meet.
- To assign a job for P_i, start from the top of the i-th line to the bottom, and make a turn whenever a horizontal segment is encountered.



Amidakuji/Ghost Leg Drawing

It is a scheme for assigning n people $P_1, \ldots, P_n$ to n jobs $J_1, \ldots, J_n$ "randomly".

- Draw vertical lines from P_i to J_i from $i = 1, \ldots, n$.
- Draw some horizontal line segments randomly between any two vertical lines that are next to each other so that no horizontal lines meet.
- To assign a job for P_i, start from the top of the i-th line to the bottom, and make a turn whenever a horizontal segment is encountered.



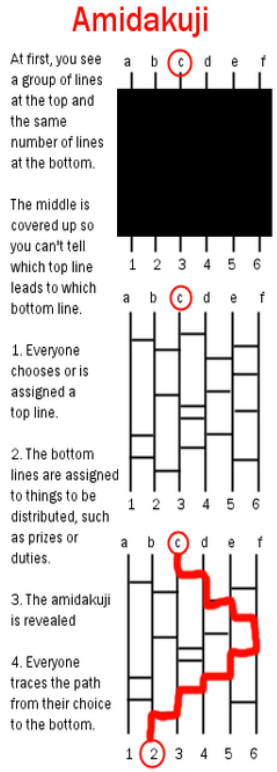
Questions

- Why do we always get an one-one correspondence (bijection)?

Amidakuji/Ghost Leg Drawing

It is a scheme for assigning n people $P_1, \ldots, P_n$ to n jobs $J_1, \ldots, J_n$ "randomly".

- Draw vertical lines from P_i to J_i from $i = 1, \ldots, n$.
- Draw some horizontal line segments randomly between any two vertical lines that are next to each other so that no horizontal lines meet.
- To assign a job for P_i, start from the top of the i-th line to the bottom, and make a turn whenever a horizontal segment is encountered.



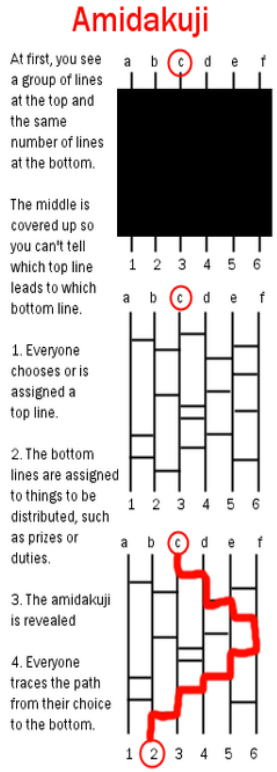
Questions

- Why do we always get an one-one correspondence (bijection)?
- Can we get all possible job assignments?

Amidakuji/Ghost Leg Drawing

It is a scheme for assigning n people $P_1, \ldots, P_n$ to n jobs $J_1, \ldots, J_n$ "randomly".

- Draw vertical lines from P_i to J_i from $i = 1, \ldots, n$.
- Draw some horizontal line segments randomly between any two vertical lines that are next to each other so that no horizontal lines meet.
- To assign a job for P_i, start from the top of the i-th line to the bottom, and make a turn whenever a horizontal segment is encountered.



Questions

- Why do we always get an one-one correspondence (bijection)?
- Can we get all possible job assignments?
- What is the minimum number of horizontal segments needed for a given job assignment?

Answer of Question 1

George Polya (1887-1985)

If one cannot solve a problem,
one can try to solve an easier problem first.

Answer of Question 1

George Polya (1887-1985)

If one cannot solve a problem,
one can try to solve an easier problem first.

- What if there is no horizontal line segment?

Answer of Question 1

George Polya (1887-1985)

If one cannot solve a problem,
one can try to solve an easier problem first.

- What if there is no horizontal line segment?
- What if there is one horizontal line segment?

Answer of Question 1

George Polya (1887-1985)

If one cannot solve a problem,
one can try to solve an easier problem first.

- What if there is no horizontal line segment?
- What if there is one horizontal line segment?
- An easy induction argument!

Bubble sort

- Regard the job assignment as a permutation (a seat reassignment)

$$\sigma = [i_1, \ldots, i_n] = \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}.$$

Bubble sort

- Regard the job assignment as a permutation (a seat reassignment)

$$\sigma = [i_1, \ldots, i_n] = \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}.$$

- Use Coxeter transpositions $(i, i+1)$ for $i = 1, \ldots, n-1$.

Bubble sort

- Regard the job assignment as a permutation (a seat reassignment)

$$\sigma = [i_1, \ldots, i_n] = \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}.$$

- Use Coxeter transpositions $(i, i+1)$ for $i = 1, \ldots, n-1$.
- For any σ, we can determine the number $\iota(\sigma)$ of inversions of σ.

Bubble sort

- Regard the job assignment as a permutation (a seat reassignment)

$$\sigma = [i_1, \ldots, i_n] = \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}.$$

- Use Coxeter transpositions $(i, i+1)$ for $i = 1, \ldots, n-1$.
- For any σ, we can determine the number $\iota(\sigma)$ of inversions of σ.
- It is the minimum number of Coxeter transpositions needed to generate σ.

Bubble sort

- Regard the job assignment as a permutation (a seat reassignment)

$$\sigma = [i_1, \ldots, i_n] = \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}.$$

- Use Coxeter transpositions $(i, i+1)$ for $i = 1, \ldots, n-1$.
- For any σ, we can determine the number $\iota(\sigma)$ of inversions of σ.
- It is the minimum number of Coxeter transpositions needed to generate σ.

Example For $\sigma = [5, 3, 1, 2, 4]$, total number of inversions is: $4 + 0 + 2 = 6$, and

$$\sigma \to [3, 5, 1, 2, 4] \to [3, 1, 5, 2, 4] \to [3, 1, 2, 5, 4]$$
$$\to [3, 1, 2, 4, 5] \to [1, 3, 2, 4, 5] \to [1, 2, 3, 4, 5],$$

Bubble sort

- Regard the job assignment as a permutation (a seat reassignment)

$$\sigma = [i_1, \ldots, i_n] = \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}.$$

- Use Coxeter transpositions $(i, i+1)$ for $i = 1, \ldots, n-1$.
- For any σ, we can determine the number $\iota(\sigma)$ of inversions of σ.
- It is the minimum number of Coxeter transpositions needed to generate σ.

Example For $\sigma = [5, 3, 1, 2, 4]$, total number of inversions is: $4 + 0 + 2 = 6$, and

$$\sigma \to [3, 5, 1, 2, 4] \to [3, 1, 5, 2, 4] \to [3, 1, 2, 5, 4]$$
$$\to [3, 1, 2, 4, 5] \to [1, 3, 2, 4, 5] \to [1, 2, 3, 4, 5],$$

So $\sigma = (1, 2)(2, 3)(3, 4)(4, 5)(1, 2)(2, 3)$.

Bubble sort

- Regard the job assignment as a permutation (a seat reassignment)

$$\sigma = [i_1, \ldots, i_n] = \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}.$$

- Use Coxeter transpositions $(i, i+1)$ for $i = 1, \ldots, n-1$.
- For any σ, we can determine the number $\iota(\sigma)$ of inversions of σ.
- It is the minimum number of Coxeter transpositions needed to generate σ.

Example For $\sigma = [5, 3, 1, 2, 4]$, total number of inversions is: $4 + 0 + 2 = 6$, and

$$\sigma \to [3, 5, 1, 2, 4] \to [3, 1, 5, 2, 4] \to [3, 1, 2, 5, 4]$$
$$\to [3, 1, 2, 4, 5] \to [1, 3, 2, 4, 5] \to [1, 2, 3, 4, 5],$$

So $\sigma = (1, 2)(2, 3)(3, 4)(4, 5)(1, 2)(2, 3)$.

Answers of Questions 2 and 3

- We can always convert a permutation σ to $[1, \ldots, n]$ using $\iota(\sigma)$ steps, where $\iota(\sigma)$ is the number of inversions of σ.

Bubble sort

- Regard the job assignment as a permutation (a seat reassignment)

$$\sigma = [i_1, \ldots, i_n] = \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}.$$

- Use Coxeter transpositions $(i, i+1)$ for $i = 1, \ldots, n-1$.
- For any σ, we can determine the number $\iota(\sigma)$ of inversions of σ.
- It is the minimum number of Coxeter transpositions needed to generate σ.

Example For $\sigma = [5, 3, 1, 2, 4]$, total number of inversions is: $4 + 0 + 2 = 6$, and

$$\sigma \to [3, 5, 1, 2, 4] \to [3, 1, 5, 2, 4] \to [3, 1, 2, 5, 4]$$
$$\to [3, 1, 2, 4, 5] \to [1, 3, 2, 4, 5] \to [1, 2, 3, 4, 5],$$

So $\sigma = (1, 2)(2, 3)(3, 4)(4, 5)(1, 2)(2, 3)$.

Answers of Questions 2 and 3

- We can always convert a permutation σ to $[1, \ldots, n]$ using $\iota(\sigma)$ steps, where $\iota(\sigma)$ is the number of inversions of σ.
- **Worst case** occurs at $[n, n-1, \ldots, 1]$; which requires

Bubble sort

- Regard the job assignment as a permutation (a seat reassignment)

$$\sigma = [i_1, \ldots, i_n] = \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}.$$

- Use Coxeter transpositions $(i, i+1)$ for $i = 1, \ldots, n-1$.
- For any σ, we can determine the number $\iota(\sigma)$ of inversions of σ.
- It is the minimum number of Coxeter transpositions needed to generate σ.

Example For $\sigma = [5, 3, 1, 2, 4]$, total number of inversions is: $4 + 0 + 2 = 6$, and

$$\sigma \to [3, 5, 1, 2, 4] \to [3, 1, 5, 2, 4] \to [3, 1, 2, 5, 4]$$
$$\to [3, 1, 2, 4, 5] \to [1, 3, 2, 4, 5] \to [1, 2, 3, 4, 5],$$

So $\sigma = (1, 2)(2, 3)(3, 4)(4, 5)(1, 2)(2, 3)$.

Answers of Questions 2 and 3

- We can always convert a permutation σ to $[1, \ldots, n]$ using $\iota(\sigma)$ steps, where $\iota(\sigma)$ is the number of inversions of σ.
- **Worst case** occurs at $[n, n-1, \ldots, 1]$; which requires

$$(n-1) + \cdots + 1 = n(n-1)/2 \text{ steps.}$$

A variation of Amidakuji

- What if we consider transpositions of the forms $(i, i+1)$ and $(i, i+2)$?

A variation of Amidakuji

- What if we consider transpositions of the forms $(i, i+1)$ and $(i, i+2)$?
- How about using transpositions $(i, i+1), (i, i+2), (i, i+3)$, etc.?

A variation of Amidakuji

- What if we consider transpositions of the forms $(i, i+1)$ and $(i, i+2)$?
- How about using transpositions $(i, i+1), (i, i+2), (i, i+3)$, etc.?

An extreme case: Using all (i, j) with $1 \leq j \leq n$

Decompose σ as product of k disjoint cycles (including fixed points).

A variation of Amidakuji

- What if we consider transpositions of the forms $(i, i+1)$ and $(i, i+2)$?
- How about using transpositions $(i, i+1), (i, i+2), (i, i+3)$, etc.?

An extreme case: Using all (i, j) with $1 \leq j \leq n$

Decompose σ as product of k disjoint cycles (including fixed points).

Then σ is a product of $n - k$ transpositions.

A variation of Amidakuji

- What if we consider transpositions of the forms $(i, i+1)$ and $(i, i+2)$?
- How about using transpositions $(i, i+1), (i, i+2), (i, i+3)$, etc.?

An extreme case: Using all (i, j) with $1 \leq j \leq n$

Decompose σ as product of k disjoint cycles (including fixed points).

Then σ is a product of $n - k$ transpositions.

So, the worst case requires $n - 1$ steps.

A variation of Amidakuji

- What if we consider transpositions of the forms $(i, i+1)$ and $(i, i+2)$?
- How about using transpositions $(i, i+1), (i, i+2), (i, i+3)$, etc.?

An extreme case: Using all (i, j) with $1 \leq j \leq n$

Decompose σ as product of k disjoint cycles (including fixed points).

Then σ is a product of $n - k$ transpositions.

So, the worst case requires $n - 1$ steps.

Example. $\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 3 & 4 & 5 & 9 & 6 & 7 & 1 & 8 & 2 \end{pmatrix} = (1, 3, 5, 6, 7)(2, 4, 9)(8).$

A variation of Amidakuji

- What if we consider transpositions of the forms $(i, i+1)$ and $(i, i+2)$?
- How about using transpositions $(i, i+1), (i, i+2), (i, i+3)$, etc.?

An extreme case: Using all (i, j) with $1 \leq j \leq n$

Decompose σ as product of k disjoint cycles (including fixed points).

Then σ is a product of $n - k$ transpositions.

So, the worst case requires $n - 1$ steps.

Example. $\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 3 & 4 & 5 & 9 & 6 & 7 & 1 & 8 & 2 \end{pmatrix} = (1, 3, 5, 6, 7)(2, 4, 9)(8).$

Then $\sigma = (1, 7)(1, 6)(1, 5)(1, 3)(2, 9)(2, 4)$.

Some open problems

Let $1 \leq m < n$, and let G_m be the set of transpositions of the form $(i, i+\ell)$ with $1 \leq \ell \leq m$.

Some open problems

Let $1 \leq m < n$, and let G_m be the set of transpositions of the form $(i, i+\ell)$ with $1 \leq \ell \leq m$.

- For a given $\sigma \in S_n$, find the smallest r such that σ is the product of r transpositions in G_m.

Some open problems

Let $1 \leq m < n$, and let G_m be the set of transpositions of the form $(i, i+\ell)$ with $1 \leq \ell \leq m$.

- For a given $\sigma \in S_n$, find the smallest r such that σ is the product of r transpositions in G_m.
- Determine the optimal (smallest) $r^* = r^*(n, m)$ so that every $\sigma \in S_n$ is a product at most r^* transpositions in G_m.

Some open problems

Let $1 \leq m < n$, and let G_m be the set of transpositions of the form $(i, i+\ell)$ with $1 \leq \ell \leq m$.

- For a given $\sigma \in S_n$, find the smallest r such that σ is the product of r transpositions in G_m.
- Determine the optimal (smallest) $r^* = r^*(n, m)$ so that every $\sigma \in S_n$ is a product at most r^* transpositions in G_m.
- To find r^* and the permutation which is most difficult to get restore, we use the breadth first search.

Partial results of the general problem

We have the following list for $r^*(n,m)$ for S_n and $(i, i+\ell)$ with $\ell \le m$,

$n\backslash m$	1	2	3	4	5	6	7	8	9	10	11
2	1										
3	3	2									
4	6	4	3								
5	10	5	5	4							
6	15	[7]	6	6	5						
7	21	[10]	8	7	7	6					
8	28	[14]	[10]	9	8	8	7				
9	36	[16]	[11]	10	10	9	9	8			
10	45	[19]	[14]	[12]	11	11	10	10	9		
11	55	[23]	[16]	[14]	13	12	12	11	11	10	
12	66	29^*	20^*	17^*	16^*	14	13	13	12	12	11

where the entries marked by brackets are obtained by computer programming.

Cayley (directed) graphs

- The Cayley digraph G generated by $T \subseteq S_n$:
  $\sigma \in S_n$ are vertices; (σ_1, σ_2) is an edge if $\sigma_2 = \tau\sigma_1$ for some $\tau \in T$.

Cayley (directed) graphs

- The Cayley digraph G generated by $T \subseteq S_n$:
  $\sigma \in S_n$ are vertices; (σ_1, σ_2) is an edge if $\sigma_2 = \tau\sigma_1$ for some $\tau \in T$.
- Find (1) the shortest path between σ and the identity $\varepsilon = [1, \ldots, n]$, and (2) the longest path (diameter) of G, denoted by $Diam(G)$.

Cayley (directed) graphs

- The Cayley digraph G generated by $T \subseteq S_n$:
  $\sigma \in S_n$ are vertices; (σ_1, σ_2) is an edge if $\sigma_2 = \tau\sigma_1$ for some $\tau \in T$.
- Find (1) the shortest path between σ and the identity $\varepsilon = [1, \ldots, n]$, and (2) the longest path (diameter) of G, denoted by $Diam(G)$.
- If $T = \{(i, i+1) : i = 1, \ldots, n-1\}$, then $Diam(G) = \binom{n}{2}$.

Cayley (directed) graphs

- The Cayley digraph G generated by $T \subseteq S_n$:
  $\sigma \in S_n$ are vertices; (σ_1, σ_2) is an edge if $\sigma_2 = \tau\sigma_1$ for some $\tau \in T$.
- Find (1) the shortest path between σ and the identity $\varepsilon = [1, \ldots, n]$, and (2) the longest path (diameter) of G, denoted by $Diam(G)$.
- If $T = \{(i, i+1) : i = 1, \ldots, n-1\}$, then $Diam(G) = \binom{n}{2}$.
- If $T = \{(i, j) : 1 \leq i < j \leq n\}$, then $Diam(G) = n - 1$.

Some open problems

Some open problems

- Open problem If $T = \{L, S\}$ with $L = (1, 2, \ldots, n)$ and $S = (1, 2)$, then the diameter is ???

S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
1	2	6	11	18	25	35	45	58	71	???

Some open problems

- Open problem If $T = \{L, S\}$ with $L = (1, 2, \ldots, n)$ and $S = (1, 2)$, then the diameter is ???

S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
1	2	6	11	18	25	35	45	58	71	???

- Conjecture If $T = \{L, S, L^{-1}\}$, then the diameter is $\binom{n}{2}$ for $n \geq 4$, attained by the path joining $[2, 1, n, n-1, \ldots, 3]$ to the identity:

S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}
1	2	6	10	15	21	28	36	45	???

Related research

- Theoretical computer science?

Related research

- Theoretical computer science? Determine the optimal sorting algorithm with the given operations, and determine the worst scenario.

Related research

- Theoretical computer science? Determine the optimal sorting algorithm with the given operations, and determine the worst scenario.
- The study of genomics and mutations,

Related research

- Theoretical computer science? Determine the optimal sorting algorithm with the given operations, and determine the worst scenario.
- The study of genomics and mutations, i.e.,
  the change of genetic sequences $x_1x_2x_3\cdots$, with $x_i \in \{A, U, G, C\}$.

Related research

- Theoretical computer science? Determine the optimal sorting algorithm with the given operations, and determine the worst scenario.
- The study of genomics and mutations, i.e.,
  the change of genetic sequences $x_1x_2x_3\cdots$, with $x_i \in \{A, U, G, C\}$.
- Quantum computing.
  It is of interest to decompose certain quantum gates into simpler quantum gates (CNOT gates).

Related research

- Theoretical computer science? Determine the optimal sorting algorithm with the given operations, and determine the worst scenario.
- The study of genomics and mutations, i.e.,
  the change of genetic sequences $x_1x_2x_3\cdots$, with $x_i \in \{A, U, G, C\}$.
- Quantum computing.
  It is of interest to decompose certain quantum gates into simpler quantum gates (CNOT gates).

Zejun Huang, Chi-Kwong Li, Sharon H. Li, Nung-Sing Sze, Factorization of permutations, preprint, arXiv:1303.3776.

Let me know if you have any questions and thoughts!

Let me know if you have any questions and thoughts!

Thank you for your attention!