COMP2521 : Assignment-2 (21T3)

How to Get Started, Part-3:

Hybrid/Meta Search Engine using Rank Aggregation

- Let T1 and T2 are two rank lists.

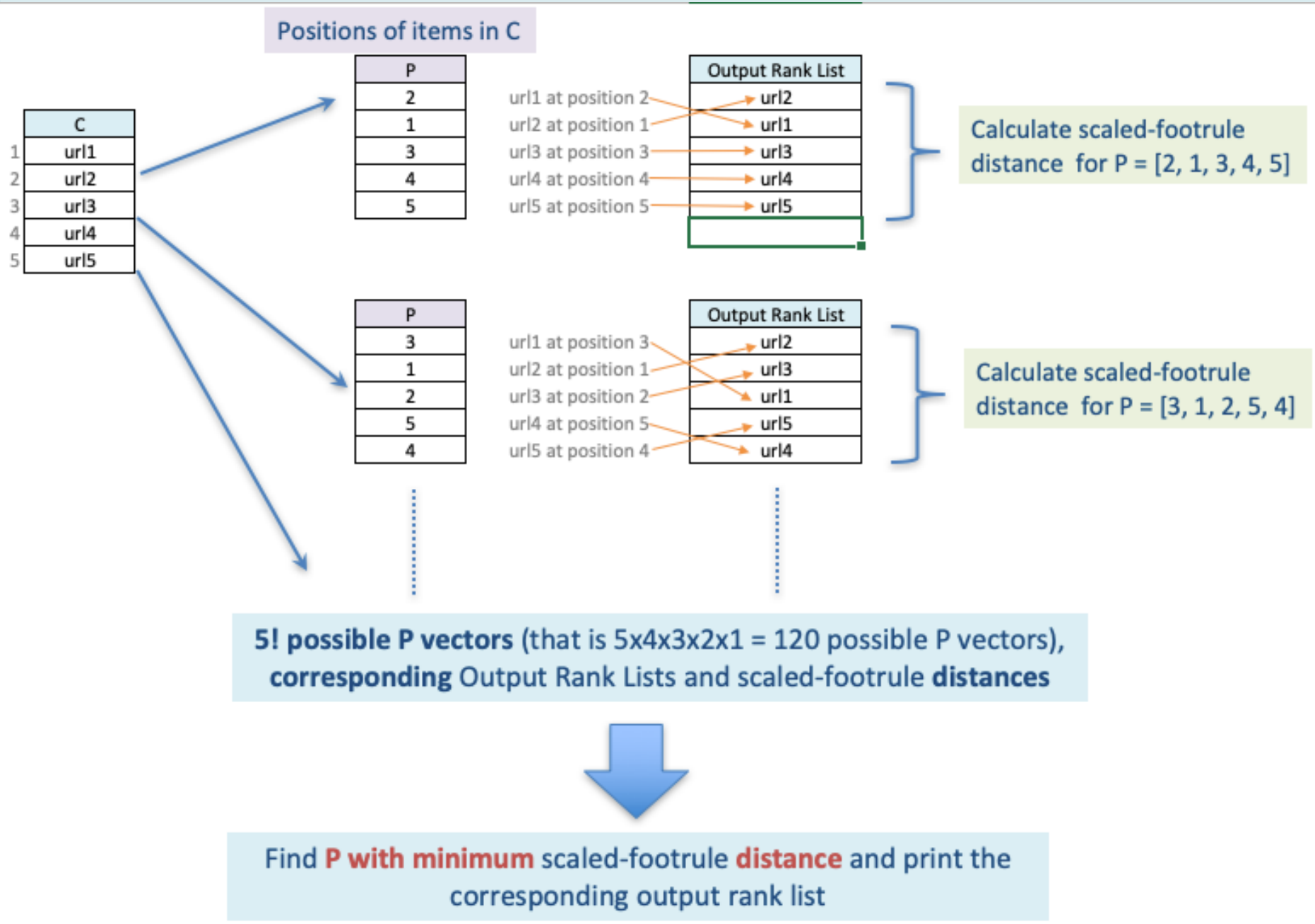
	size of T1 is 5	*size of T2 is 4*
	T1	T2
1	url1	url3
2	url3	url2
3	url5	url1
4	url4	url4
5	url2	

- Let C = set of nodes to be ranked (**union** of T1 and T2)

C
url1
url2
url3
url4
url5

- From C, we can generate many possible output rank lists by changing the order of items in C (url1, url2, etc.). If we have say N items, there are N! ways to generate such output rank lists (**see the next slide**).
- A very **simple** and obviously inefficient **approach** could use brute-force search
  - generate all possible alternatives,
  - calculate scaled-footrule distance for each alternative, and
  - find the alternative with minimum scaled-footrule distance.

- For example, see the next slide ….


Positions of items in C
C
1 url1
2 url2
3 url3
4 url4
5 url5
P
2
1
3
4
5
url1 at position 2
url2 at position 1
url3 at position 3
url4 at position 4
url5 at position 5
Output Rank List
url2
url1
url3
url4
url5
Calculate scaled-footrule distance for P = [2, 1, 3, 4, 5]
P
3
1
2
5
4
url1 at position 3
url2 at position 1
url3 at position 2
url4 at position 5
url5 at position 4
Output Rank List
url2
url3
url1
url5
url4
Calculate scaled-footrule distance for P = [3, 1, 2, 5, 4]
5! possible P vectors (that is 5x4x3x2x1 = 120 possible P vectors), corresponding Output Rank Lists and scaled-footrule distances
Find P with minimum scaled-footrule distance and print the corresponding output rank list

How to Calculate scaled-footrule for a P vector

size of T1 is 5 *size of T2 is 4*

	T1	T2
1	url1	url3
2	url3	url2
3	url5	url1
4	url4	url4
5	url2	

n is 5

C	P	W(C,P) for T1	W(C,P) for T2
url1	1	abs(1/5 - 1/5)	abs(3/4 - 1/5)
url2	3	abs(5/5 - 3/5)	abs(2/4 - 3/5)
url3	2	abs(2/5 - 2/5)	abs(1/4 - 2/5)
url4	5	abs(4/5 - 5/5)	abs(4/4 - 5/5)
url5	4	abs(3/5 - 4/5)	

p

n (5)

size of T1 (5)

T_1(url5) - position of url5 in T1, that is 3

p

n (5)

size of T2 (4)

T_2(url3) - position of url3 in T2, that is 1

$$W(c,p) = \sum_{i=1}^{k} \left| \tau_i(c)/|\tau_i| - p/n \right|$$

W(C,P) is sum of all yellow cells
(1.6 in the above example)

"smart" Algorithm

- If you use a simple brute-force search, you will receive 65% of the maximum marks for Part-3.
- However, you will be rewarded up 100% for part-3 if you implement a "smart" algorithm that avoids generating unnecessary alternatives, in the process of finding the minimum scaled-footrule distance.
- Please document your algorithm such that your tutor can easily understand your logic, and clearly outline how you plan to reduce search space, otherwise you will not be awarded mark for your "smart" algorithm!
- Yes, it's only few marks, but if you try it, you will find it very challenging and rewarding.