1. 제 28회 전국 ICT창의성대회 프로그래밍전문가 부문 금상 / 2017 서강대학교 총장배 전국 고등학생 알고리즘 경진대회 은상

Kibwa

제17-022호

상 장

프로그래밍전문가 부문 선린인터넷고등학교 3학년

금 상 권 욱 제

위 사람은 과학기술정보통신부가 후원하고 전국컴퓨터교육협의회가 주최한 제 28회 전국 ICT창의성대회에서 위와 같이 우수한 성적으로 입상하였기에 상장을 수여합니다.

2017년 8월 25일

사단법인 IT여성기업인협회 회장 장 혜 원

제 5 호

상 장

융합기초교육원장상

학교 : 선린인터넷고등학교
학년 : 3학년
성명 : 권 욱 제

위 사람은 서강대학교 가 주최한 2017 『서강대학교 총장배 전국 고등학생 알고리즘 경진대회』에서 우수한 성적으로 위와 같이 입상하였기에 이에 상장을 수여함.

Be as proud of Sogang
as Sogang is proud of you

2017 년 08 월 26 일

서강대학교 총장 박 종 구

2. 한국정보과학회 프로그래밍언어연구회 겨울학교 (SIGPL Winter School 2017) 참여

한국정보과학회
KOREAN INSTITUTE OF INFORMATION SCIENTISTS AND ENGINEERS

[06704] 서울특별시 서초구 방배로 76(방배동, 머리재빌딩 401호) T.1588-2728 F.02-521-1352 E-mail.kiise@kiise.or.kr

제17−250호

학술행사 참가확인서

※ 성　　명 : 권욱제
※ 소　　속 : 선린인터넷고등학교

위 사람은 본 학회 프로그래밍언어연구회에서 주최한 아래 학술행사에 참가하였음을 확인함.

= 아　　　래 =

- ❑ 행사명 : 프로그래밍언어연구회 겨울학교(SIGPL Winter School 2017)
- ❑ 일자 및 장소 : 2017년 2월 8일 ~ 10일, KAIST

2017년 2월 13일

한국정보과학회 회장

O(n log n) algorithm for the restriction scaffold problem

Wookje Kwon (wookje.happy@gmail.com)
Sunrin Internet High Shcool
23/7/2017
(Abridgement)

1 Introduction

The problem takes two finite sets and T with $S|>|T|$. For given sets of numbers, let a matching of S and T be a set of pairs (s,t), s S, $t\in T$, such that each number in S is paired with at least one number in T and each number in T is paired with at least one number in S. The cost of matching of (s,t) is $|s-t|$. In this paper, we present $O(n\log n)$ algorithm for this problem, unlike existing algorithms.

2 Preliminaries

Here are the example sets of the restriction scaffold problem below.

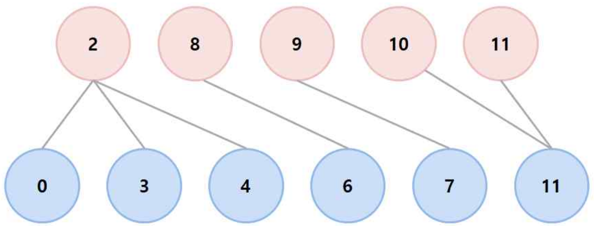

Figure 1. Example connection of sets $\{2,8,9,10,11\}$ and $T=\{0,3,4,6,7,11\}$

For any finite set, such A, let number of elements belonging to A be $N(A)$. Let n be $N(S)$ and m be $N(T)$.

Lemma 1 *If there exists optimal connection* (s,t_1) *and* (s_2,t_2) *with* $s_1<s_2$, $t_1>t_2$, *then there exists optimal connection* (s_1,t_2) *and* (s_2,t_1).

Proof *Lemma 1* can be easily deducted from the fact that the other elements are not affected and the cost does not increase by changing the pair connection. Therefore, we can deduct $O(nm)$ algorithm by D_{ij}. Define D_{ij} as *the minimum cost when* i *and* j *in ascending order in* S *and* T *satisfy the condition of connection*$D_{ij}=\min(D_{i-1,j},D_{i,j-1},D_{i-1,j-1})+|S_i-T_j|$, and D_{ij}. In this way, we can solve the problem naively.

Lemma 2 *If there exists optimal assignment* (s_1,t_1) *and* (s_2,t_2) *with* $s_1<s_2$, $t_1<t_2$, *then there exists optimal assignment that connect neither* (s_1,t_2) *nor* (s_2,t_1).

Proof *Lemma 2* is shown as following.

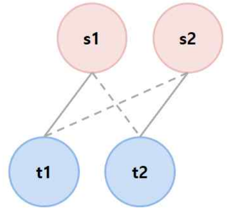

Figure 2. Figure of lemma 2

That is, if (s_1,t_1) and (s_2,t_2) are connected as shown in the *figure 2*, connecting (s_1,t_2) or (s_2,t_1) is not beneficial. Thanks to these lemmas, we can divide all connections into two types.

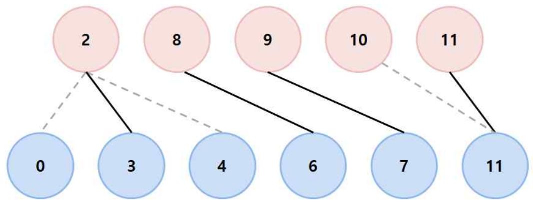

Figure 3. Divide *figure 1* into two connections.

Let two connections, strong connection and week connection. A strong connection connects the S and T elements one by one, while a weak connection connects the S or T element to the nearest other element. This leads to a $O(n+m)$ solution.

3 Computing a minimum cost assignment

Sort the S and T elements together in ascending order, and define D_i be *minimum cost when the condition of connection is satisfied up to the* i*-th element*. We can find that there are two cases.

The first case is that the i-th element is connected to a weak connection. It can be easily compute in $O(1)$.

The second case is when $j+1$ to i-th correspond to a strong connection by one-to-one for any j. In this case, $N(S)$ and $N(T)$ must be the same among the $j+1$ to i-th. There can be multiple such j, and it can be seen that only the largest j is sufficient. For smaller j', let it be optimal to correspond one-to-one to $j'+1$ to i-th. D_j implicates the case $j'+1$ to j-th and D_i does the case $j+1$ to i-th, therefore, it is sufficient to consider only the largest j.

But it seems difficult to compute. Since it is the largest j that the $N(S)$ and $N(T)$ among $j+1$ to i-th are same, in $j+1$ to $i-1$, $N(S)$ is always bigger than $N(T)$, or vice versa. Thus, the minimum cost corresponding to one-to-one correspondence with $j+1$ to i-th elements is $|N(S)-N(T)|$.