Karakterden Bağımsız Birçok Dile Uyarlanabilir Dizin Sıralama Algoritması
Applicable Index Sorting Algorithm Regardless of Character For Many Languages

[1]Burhan BARAKLI and [1]Ahmet KÜÇÜKER
[1]Faculty of Engineering, Department of Electrical and Electronics Engineering Sakarya University, Turkey

Abstract:

There were various sorting algorithm used in alphabetical sorting methods at index creation stage. Generally this sorting algorithms includes alphabetical sorting methods and some problems occurs because of character order differences in various language alphabets and fonts. Especially problems occurs by indexing with traditional sorting algorithms which fonts used in dialect studies in Turkish Language and Literature Department designed by Turkish Language researchers. The most important of the problems encountered in indexing studies is disorder of words that begin with the special character of fonts in index. In this study applicable index sorting algorithm regardless of character for many languages and user defined index sorting aimed.

Key words: Indexing, alphabetical sorting, character regardless

Özet:

Dizin oluşturulma aşamasında alfabetik sıralama yöntemlerinde kullanılan çeşitli sıralama algoritmaları mevcuttur. Bu sıralama algoritmaları genellikle alfabetik sıralanma yöntemini içermektedir ve çeşitli dillerde kullanılan alfabelerdeki ve yazı tiplerindeki karakter sıralamaları farklılıklarından dolayı problemler oluşturmaktadır. Özellikle Türk Dili ve Edebiyatı bölümlerinde ağız çalışmalarında kullanılmakta olan fontların büyük bir kısmı Türk dili araştırmacıları tarafından tasarlanmış olup kullanılan geleneksel sıralama algoritmaları ile dizin oluşturulmasında sorunlar ortaya çıkmaktadır. Dizin çalışmalarında karşılaşılan sorunlardan en önemlisi yazı tiplerinde yer alan özel karakterlerle başlayan kelimelerin dizindeki yerlerinde meydana gelen karışıklıklardır. Bu çalışmada çok dilde kullanılabilir ve alfabetik sıralaması ile karakterlerin tanımlanması kullanıcı tarafından değiştirilebilir bir sıralama algoritması gerçekleştirilmesi ve belirlenen sıralamada dizin oluşturulması hedeflenmiştir.

Anahtar Kelimeler: Dizin oluşturma, alfabetik sıralama, karakter bağımsız,

1. Giriş

Bilimin hızlı gelişmesi ve insanların hep daha iyiyi istemeleri, kullanılan araçların gelişmesine neden olmaktadır. Bilgisayar sistemlerinde de benzer bir durum söz konusudur. Teorik yapılar ve testler çok güçlü bilgisayar ve gömülü sistemlere ihtiyaç duymaktadır. Kullanılan donanımın arttırılması, işlemci hızını veya bellek miktarını arttırmak gibi çözümler işlem hızını belirli bir oranda arttırsa da yine de istenilen hıza veya kapasiteye ulaşılamamaktadır. Aynı zamanda, bu gibi çözümler fiziksel yetersizlerden dolayı sınırlı seviyede kalmaktadır. Örneğin, video işleme için geliştirilen kodek yapısı MPEG-2, teorik yapısının ispatından ancak 10 yıl sonra donanım olarak hazırlanabilmiştir. Bu durumların çözümü için bir takım algoritmalar sunulmuştur [1-8].

*Corresponding author: Address: Faculty of Engineering, Department of Electrical and Electronics Engineering Sakarya University, 54187, Sakarya TURKEY. E-mail address: barakli@sakarya.edu.tr, Phone: +902642955591

Sıralama algoritması, bilgisayar bilimlerinde ya da matematikte kullanılan, verilen bir listenin elemanlarını belirli bir sıraya sokan algoritmadır. Bu algoritmalar bilgisayar sistemlerini en çok zorlayan algoritmaların başında gelmektedir. Düşük boyutlu dizilerin sıralanmasında problem olmasa da, büyük boyutlu dizilerin sıralanmasında sorunlar oluşmakta ve uzunca süreler sistemleri meşgul etmektedir. En çok kullanılan sıralama türleri, sayı sıralama ve alfabetik sıralamadır. Arama işlemini gerçekleştiren algoritmaların başarımının yüksek olması için sıralama işleminin verimli yapılması gerekmektedir. Ayrıca bilgisayarda tutulan verilen düzenlemesi ve kullanıcının daha rahat belge vs. gibi dosyaları kullanabilmesini sağlamaktadır.

Sıralama algoritmaları, yapacağı işlev basit olsa da, çözümü karmaşık olan bir işi gerçekleştirdikleri için, üzerinde fazlaca araştırma yapılan bir konu olmaktadır. Literatürde birçok sayı ve karakter sıralama işlemini yapan algoritma bulunmaktadır [6]. Bu çalışmada ise karakterden bağımsız olmak üzere istenilen şekilde sıralama yapan bir algoritma sunulmuştur.

Sıralama algoritmaları genellikle hesaplama karmaşıklığı, yer değiştirme karmaşıklığı, bellek kullanımı, özyineleme ve kararlılık ölçütlerine göre sınıflandırılır. Hesaplama karmaşıklığı, dizideki öğelerin karşılaştırılmasının en iyi, ortalama ve en kötü başarımının dizinin boyutu cinsinden gösterilmiş halidir. Yerinde sıralama algoritmaları için yer değiştirme karmaşıklığı ölçüttür. Bazı sıralama algoritmaları dizinin içerdiği öğelerin dizinin saklandığı alanda sıralar. Bazı algoritmalar ya özyinelemeli ya da özyinelemesiz çalışırken, birleştirmeli sıralama gibi bazı algoritmalar iki biçimde de uygulanabilir.

Literatürdeki en çok kullanılan sıralama algoritmaları ağaç sıralaması, birleştirmeli sıralama, saçma sıralama, sayarak sıralama, hızlı sıralama, kabarcık sıralaması, kabuk sıralaması, kokteyl sıralaması, kova sıralaması, basamağa göre sıralama, rahat sıralama, seçmeli sıralama, tarak sıralaması, topolojik sıralama, yığın sıralaması gibi birçok sıralama algoritması mevcuttur [6].

Literatürdeki karakter sıralama ile ilgili kullanılan algoritma basamağa göre sıralama (BGS) algoritmasıdır [9]. Ancak BGS ile standart bir sıralama söz konusudur. Keyfi bir sıralama gerçekleştirilmemektedir. Konunun ve problemin hızlı bir şekilde anlaşılır olması açısından şu şekilde bir örnek verilebilir: geleneksel karakter sıralama algoritmaları A-B-C-D-…-V-Y-Z'ye kadar sıralama işlemini gerçekleştirmektedirler ancak G-A-T-Z-…-C-B şeklinde olması istenilen bir sıralama için özelleştirilebilir bir sıralama algoritmasına ihtiyaç duyulmaktadır. Bilgisayar sistemlerinde geleneksel sıralama algoritmaları kullanılmakta ve son kullanıcıya sıralamayı değiştirme hakkı vermemektedir. Bu çalışmanın amacı ise son kullanıcının isteğine göre sıralama yapan bir yöntem sunmaktır.

Sunulan yöntem geleneksel karakter sıralama algoritmalarından farklı olarak sayısal bir temel alınarak sıralama işlemini gerçekleştirmektedir. Herhangi bir sayısal değerleri sıralama algoritması ile basamak değerlendirme olarak sunulan yöntemin birleştirilmesiyle yeni bir alfabetik sıralama algoritması gerçekleştirilmiştir. Çalışmanın literatüre katkısı Türk dili ağız çalışmaları için sağlanmış, birçok tezin ve akademik çalışmanın dizin oluşturulması başarıyla gerçekleştirilmiştir [10].

2. Önbilgi

Konunun daha iyi anlaşılır olması açısından hem sayısal hem de alfabetik sıralama örnekleri verilecek ve sonrasında önerilen algoritma sunulacaktır. Basit ve anlaşılır olması bakımından sayısal değerleri sıralama algoritması olarak kabarcık sıralama ve basamağa göre sıralama bir sonraki kısımda tartışılacaktır.

2.1. Kabarcık Sıralama

Sıralama algoritmaları içerisindeki en yalın algoritma kabarcık sıralamasıdır. Sıralanacak dizi içinde ilerlenirken her bir işlemde iki elemanın karşılaştırılıp, karşılaştırılan elemanların yanlış sırada olmaları durumunda yerleri değiştirilmesi mantığına dayanan bir algoritmadır. Algoritma, herhangi bir değişiklik olmayan döngü sayısına kadar dizinin başına dönerek kendisini tekrar eder.

Birinci Döngü:
[6 2 5 3 9] >>> [2 6 5 3 9] İlk iki eleman karşılaştırılır ve yerleri değiştirilir.
[2 6 5 3 9] >>> [2 5 6 3 9]
[2 5 6 3 9] >>> [2 5 3 6 9]
[2 5 3 6 9] >>> [2 5 3 6 9] Elemanlar sıralı olduğu için algoritma herhangi bir değişiklik yapmaz.
İkinci Döngü:
[2 5 3 6 9] >>> [2 5 3 6 9]
[2 5 3 6 9] >>> [2 3 5 6 9]
[2 3 5 6 9] >>> [2 3 5 6 9]
[2 3 5 6 9] >>> [2 3 5 6 9]

Dizi şu anda sıralıdır. Ancak algoritma elemanların sıralı olup olmadığını bilememektedir. Bu nedenle yukardaki işlemlerin son bir kez tekrarı yapılarak hiçbir değişiklik yapmayan döngü sayısına kadar işlemler son bir kez tekrarlanır. Sonuç olarak üçüncü döngü sonunda dizi sıralıdır ve algoritma sonlandırılır.

2.2. Basamağa göre sıralama algoritması

Sayma sayıları adlar veya karakterler dizilerini ASCII denen sayısal değerleri sıralama için kullanılan bir algoritmadır. Dolayısıyla basamağa göre sıralama algoritması yalnızca sayma sayılarını sıralamak için kullanılan bir algoritma değildir. Örneğin "k, l, a, ba, m, cd, t" dizisi sözlük sırasına göre "a, ba, cd, k, l, m, t" olarak sıralanacaktır. Meraklı okuyucu ilgili çalışmayı inceleyebilir [9].

3. Önerilen Yöntem

Çalışmanın amacı kısaca Türk dili çalışmalarında hem isteğe bağlı değiştirilebilen bir alfabetik sıralamasının talebi hem de font değişikliklerinden kaynaklanan alfabetik sıralamanın yanlış olmasından dolayı yeni bir algoritma arayışıdır. Herhangi bir yazılım dilinde (c, Java, Pascal) bir

sıralama fonksiyonu ya da Excel, Word gibi kelime işlemcileri, sıralamayı ASCII koduna göre yapmaktadır. Ancak istenilen sıralama her zaman bu şekilde olmayabilir. Örneğin Baha, Ali, Taha, Ahmet, Ayşe, Evren, Àbdullah, Ëlif gibi bir kümenin alfabetik sıralanması ile Ali – Ahmet – Ayşe – Baha – Evren – Taha – Àbdullah – Ëlif şeklinde olmaktadır. Çünkü À ve Ë karakterlerinin ASCII kodları uluslararası karakterlere uymadığından sıralamada sona atılmaktadır. Ancak olması gereken sıralama Ali – Àbdullah – Ahmet – Ayşe – Baha – Evren – Ëlif – Taha şeklindedir. Ayrıca son kullanıcının isteğine göre alfabetik sıralamada değişiklik yapılması istenebilir. Alfabetik sıralamanın T, M, A, D, C, N, B ….. şeklinde olduğu farz edilirse ve Abdullah, Amca, Derya, Cenk, Nalan, Taha, Melih şeklindeki bir kümenin sıralanması verilen alfabetik sıralamaya göre Taha, Melih, Amca, Abdullah, Derya, Cenk, Nalan şeklinde olmaktadır.

Yukarıdaki verilen örnekteki gibi bir sıralama için fontların ASCII değerlerine yerine her karaktere sayı değeri verilmektedir. Konunun rahat anlaşılır olması açısından bir örnek verilecektir ardından genel şablon tanıtılacaktır.

Alfabenin 12 karakterden oluştuğu ve alfabetik sıralamanın şekli Alfabe= {D, M, C, B, J, K, P, I, F, E, G, X} olduğu farz edilmiştir. Sıralama için kelimelerin basamak değerlerini hesaplayan yeni bir yöntem sunulmuştur. Örneğin $GIFJD$ gibi bir kelimenin basamak değeri aşağıdaki şekilde hesaplanır. $\varphi = 12$ basamak değeri ve $\#(x)$, x harfine ait alfabetik sıralamadaki karakter değeri olmak üzere, 12 harften oluşan bir alfabe Tablo 1'de verilmiştir.

Tablo 1. 12 harften oluşan bir alfabe

$\#(x)$ basamak değeri			
D	12	P	6
M	11	I	5
C	10	F	4
B	9	E	3
J	8	G	2
K	7	X	1

$$GIFJD = 2 \times 12^4 + 5 \times 12^3 + 4 \times 12^2 + 8 \times 12^1 + 12 \times 12^0 = 50796$$

Örnek bir dizi Tablo 2'de verilmiştir. Dizinin eleman sayısı 10'dur. Basamak değerlerinin hesaplanmasının ardından herhangi bir sayısal sıralama algoritmasının kullanılması ile dizi sıralanmış bir şekilde oluşturulabilir. Bu sayede keyfi bir alfabetik sıralama algoritması gerçekleştirilmiş olur. Basamak değerinin hesabı genel bir şablon olarak, n alfabedeki harf sayısı, K_n n. harf ve kelime uzunluğu l olmak üzere kelime basamak değeri BD,

$$BD = \sum_{p=1}^{l} \#(K_p) \, . \, n^l$$

denklemi ile hesaplanır.

Tablo 2. Verilen alfabeye ait oluşturulan kelimeler, basamak değerleri ve sıralanmış kelime dizisi

Kelime Dizisi	Basamak Değeri		Sıralanmış Dizi
DMJ	1868		DMC
KDI	1085		DMJ
GXF	304		CMD
GBJ	404		CGX
GMC	430		CX
DMC	1870		CGG
CMD	1584		KDI
CGX	1465		GMC
CX	1452		GBJ
CGG	1466		GXF

4. Sonuçlar

Sunulan algoritma ile oluşturulan bir kitap dizini Şekil 1'de verilmiştir [10]. Türk dizin çalışmalarında kullanılan değişik fontların sebep olduğu alfabetik sıralama problemleri önerilen şablon ile giderilmiştir. Yöntem uygulanırken bir veri tabanı kullanılmıştır. Sayısal sıralamayı gerçekleştirmek için hızlı sıralama algoritması yönteme eklenmiştir. Ayrıca değişik dillerdeki alfabetik sıralamada oluşan problem giderilmiştir.

ʿacāyib
ʿaceple
ʿacizlen
afsentīn
aġac
aġar
agac
agaç
aḫlāṭ
aḫżar
aḫzar

Şekil 1. Değişik fontların istenilen sırada oluşturulduğu bir sıralama dizisi

5. Tartışma

Alfabetik sıralamada kullanılmak üzere etkili, hızlı ve basit bir yöntem sunulmuştur. Dünyada birçok dil yapısı var olduğundan alfabetik sıralama için bir standart söz konusu olmayacağı açıktır. Ayrıca son kullanıcının değişik bir alfabetik sıralama isteği mevcut algoritmalar ile gerçekleştirilememektedir. Sunulan yöntem ile dil yapısından ve font değişikliğinden bağımsız olarak istenilen şekilde alfabetik sıralama başarıyla gerçekleştirilmektedir.

Referanslar

[1] Wei PZG, Lu M, Yang ZQ. Visual Basic programmer concise course of study (secondly edition), altitude Education Publishing Company, Beijing, pp. 147-152, 2003
[2] Huo YX, Ju X, Tang BJ. Practical Data Structure, Shanghai Publishing Company,pp.103-106, 2003
[3] Zhou JQ, Ma XJ. Exceeding Quick Sorting Computation. Computer Application. Vol.16,no.3,25-28, 1995Sartaj Sahni: Data Structures,Algorithms, and Applications in C++,MCGraw-hill, pp.293-297, 1999
[4] Sahni S. Data Structures,Algorithms, and Applications in C++,MCGraw-hill, pp.293-297, 1999
[5] Zhou JQ, Ma XJ. Exceeding Quick Sorting Computation. Computer Application. Vol.16,no.3,25-28, 1995
[6] Knuth DE. The Art of Computer Programming – Sorting and Searching. Addison Wesley Publishing Company, Inc., 1973, 3:145-158
[7] Iraj H., Afsari MHS. Hassanzadeh S. A New External Sorting Algorithm with Selecting the Record List Location. USEAS Transactions on Communications. 2006, 5(5):909-913.
[8] Feng H. Analysis of the Complexity of Quick Sort for two Dimension Table. Jisuanji Xuebao. In Chinese. 2007, 30(6):963-968.
[9] Maclaren MD. Internal Sorting by Radix Plus Sifting. Journal of the Association for Computing Machinery, 1966 Vol. 13(No. 3): pp.404-411.
[10] Uçar İ. Kavâidü'r-Remy "Ok Atıcılığının Kuralları" / Abdullah el-Kâtip, Mucize Yayınları, 491 s, Ankara 2013.