



**INTRODUCTION TO ARTIFICIAL  
INTELLIGENCE HOMEWORK-3**

Muhammed İzzet DEMİR-191180032

**22.11.2023**

# GİRİŞ

Bu çalışma, geniş öncelikli arama, derinlik öncelikli arama, A\* arama, IDA, Greedy Best First Search gibi popüler yapay zeka tekniklerinin performansını N-Puzzle problemi üzerinden karşılaştırmaktadır. Her algoritmanın çözüme ulaşmaya çalışırken sergilediği karmaşıklığı değerlendirmek ve farklı durumlarda hangi tekniğin daha etkili olduğunu belirlemek amacıyla gerçekleştirilmiştir. N Puzzle, test senaryosu olarak kullanılmış ve her algoritma için bir uygulama oluşturularak sonuçlar elde edilmiştir. Bu makale, her algoritmanın çözüm sürecindeki ilerleme düzeyini göstererek, her bir tekniğin tercih edilebileceği belirli durumları açıklamayı amaçlamaktadır.

## Geniş Öncelikli Arama

Bu algoritma tamamlayıcı değildir, çünkü sol alt ağaç sınırsız derinlikte olup çözüm içermiyorsa sonlanmaz. Zaman karmaşıklığı  $O(b^m)$  ve mekan karmaşıklığı  $O(b^m)$  olarak ifade edilebilir. Optimal değildir, çünkü bir hedef düğümü bulsa bile tüm sol alt ağacı keşfeder.

## Derin Öncelikli Arama

Derinlik öncelikli arama da tamamlayıcı değildir ve zaman ile mekan karmaşıklığı geniş öncelikli aramaya benzer. Optimal değildir, çünkü bir hedef düğümü bulsa bile tüm sol alt ağacı keşfeder.

## Derinlik Sınırlı Arama

Zaman karmaşıklığı  $O(b^m)$  olan bu algoritma, uygun bir kesim seçildiğinde çözüm bulmayı garanti eder. Mekan karmaşıklığı  $O(bm)$  ise en uzun yolun uzunluğu kadar olacaktır. Optimal değildir, çünkü adımların sabit olması durumunda bir çözümü garanti etmez.

## Uniform Cost Search

Geniş öncelikli arama ile benzer özelliklere sahiptir. Tamamlayıcı değildir, çünkü sol alt ağaç sınırsız derinlikte olup çözüm içermiyorsa sonlanmaz. Zaman ve mekan karmaşıklığı geniş öncelikli arama ile aynıdır.

## İteratif Derinlemesine Arama

Bu algoritma tamamlayıcıdır, ancak zaman karmaşıklığı  $O(b^d)$  ve mekan karmaşıklığı  $O(b^d)$  şeklinde ifade edilebilir. Optimal değildir, çünkü adımların sabit olması durumunda bir çözümü garanti etmez.

## Açgözlü Arama (Greedy Search)

Bu algoritma tamamlayıcı değildir ve döngülere sıkışabilir. Zaman karmaşıklığı  $O(b^m)$  olup, iyi bir sezgisel ile önemli ölçüde iyileştirilebilir. Mekan karmaşıklığı  $O(b^m)$  şeklinde ifade

edilebilir. Optimal değildir, çünkü başka olasılıkları denemeden sonsuz bir yolu takip edebilir.

**A \***

A\* arama tamamlayıcıdır, zaman karmaşıklığı üstel  $O(b^m)$  ve mekan karmaşıklığı  $O(b^m)$  şeklinde ifade edilir. Optimaldir, yani en düşük maliyetli çözümü bulma garantisi verir.

## KARŞILAŞTIRMA TABLOSU

<b>Algoritma</b>	<b>Tamamlanabilirlik</b>	<b>Zaman Karmaşıklığı</b>	<b>Mekan Karmaşıklığı</b>	<b>Optimalite</b>
Geniş Öncelikli	Hayır	$O(b^m)$	$O(b^m)$	Hayır
Derinlik Öncelikli Arama	Hayır	$O(b^m)$	$O(bm)$	Hayır
Derinlik Sınırlı Arama	Evet	$O(b^m)$	$O(bm)$	Hayır
Biriiform Maliyet Arama	Hayır	$O(b^m)$	$O(b^m)$	Hayır
İteratif Derinlemesine Arama	Evet	$O(b^d)$	$O(b^d)$	Hayır
Açgözlü Arama	Hayır	$O(b^m)$	$O(b^m)$	Hayır
A* Arama	Evet	Üstel $O(b^m)$	$O(b^m)$	Evet

## KOD ÇIKTILARIM

```

Python Debug Console
Enter N (3, 8, 15, 24, 35 ...) -> 15
initial random state:
[14 3 8 5]
[6 13 9 7]
[2 0 1 10]
[15 12 4 11]
Choose an algorithm [Breadth first(1), Depth first(2), Uniform cost(3), Depth limited(4), Iterative
deepening(5), A*(6), Greedy(7), ] -> 1

----- Output Information -----
Time taken: 1.598119
G-value (level solution found in goal tree): 413
Processed nodes: 5202
Max stored nodes: 2717
Do we find a solution: True
Do you want to try another algorithm enter or show the graffic(g) :
Choose an algorithm [Breadth first(1), Depth first(2), Uniform cost(3), Depth limited(4), Iterative
deepening(5), A*(6), Greedy(7), ] -> 2

----- Output Information -----
Time taken: 1.988462
G-value (level solution found in goal tree): 303
Processed nodes: 6870
Max stored nodes: 3589
Do we find a solution: True
Do you want to try another algorithm enter or show the graffic(g) :
Choose an algorithm [Breadth first(1), Depth first(2), Uniform cost(3), Depth limited(4), Iterative
deepening(5), A*(6), Greedy(7), ] -> 3

----- Output Information -----
Time taken: 1.790646
G-value (level solution found in goal tree): 305
Processed nodes: 5773
Max stored nodes: 3011
Do we find a solution: True
Do you want to try another algorithm enter or show the graffic(g) :
Choose an algorithm [Breadth first(1), Depth first(2), Uniform cost(3), Depth limited(4), Iterative
deepening(5), A*(6), Greedy(7), ] -> 4

```

```

----- Output Information -----
Time taken: 0.872065
G-value (level solution found in goal tree): 239
Processed nodes: 3053
Max stored nodes: 1560
Do we find a solution: True
Do you want to try another algorithm enter or show the graffic(g) :
Choose an algorithm [Breadth first(1), Depth first(2), Uniform cost(3), Depth limited(4), Iterative
deepening(5), A*(6), Greedy(7), ] -> 5

----- Output Information -----
Time taken: 1.203494
G-value (level solution found in goal tree): 191
Processed nodes: 3716
Max stored nodes: 1941
Do we find a solution: True
Do you want to try another algorithm enter or show the graffic(g) :
Choose an algorithm [Breadth first(1), Depth first(2), Uniform cost(3), Depth limited(4), Iterative
deepening(5), A*(6), Greedy(7), ] -> 6

----- Output Information -----
Time taken: 2.653768
G-value (level solution found in goal tree): 254
Processed nodes: 8933
Max stored nodes: 4669
Do we find a solution: True
Do you want to try another algorithm enter or show the graffic(g) :
Choose an algorithm [Breadth first(1), Depth first(2), Uniform cost(3), Depth limited(4), Iterative
deepening(5), A*(6), Greedy(7), ] -> 7

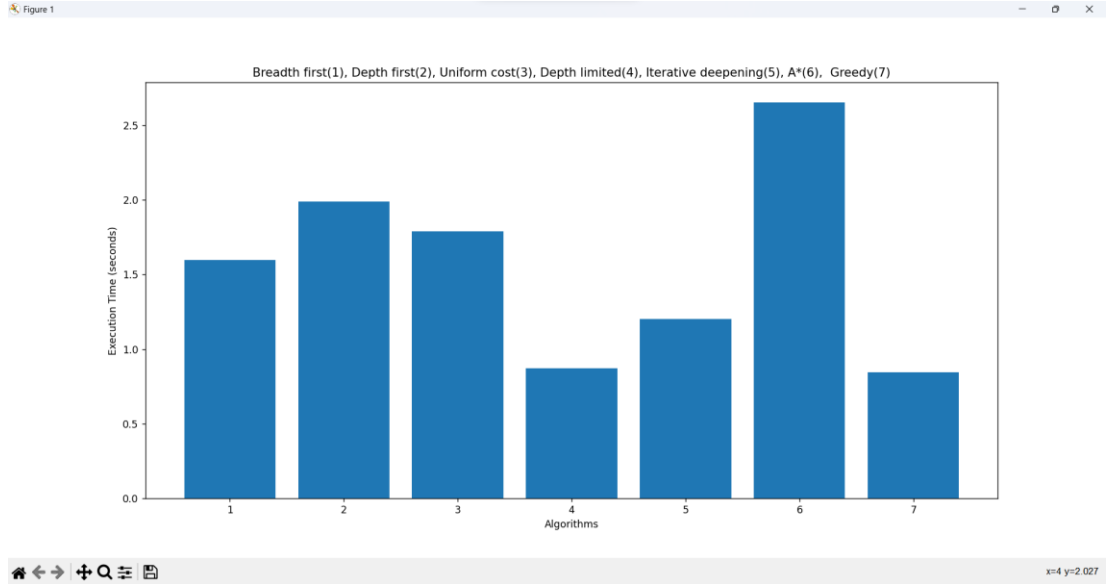
----- Output Information -----
Time taken: 0.845062
G-value (level solution found in goal tree): 170
Processed nodes: 2931
Max stored nodes: 1527
Do we find a solution: True
Do you want to try another algorithm enter or show the graffic(g) : g

```

Yukarıdaki örnek programın çıktılarına göz attığımızda, 7 farklı algoritmanın performansını değerlendirmek amacıyla gezdikleri notları içeren bilgileri görmekteyiz. Bu çıktılara dikkat ettiğimizde, programın bir 4 boyutlu matris üzerinde çalışmasına rağmen sonuçları hızlı bir şekilde üretebildiğini gözlemliyoruz. Bu hızlı çıktı üretme yeteneği, algoritmanın optimum çalışma koşullarını sağlaması ve gerekli işlemleri etkili bir şekilde gerçekleştirmesinden kaynaklanmaktadır.

Programın çalışma hızı, bilgisayarın özelliklerine bağlı olarak değişebilir. Ancak, genel olarak, ortalama bir bilgisayarda bu algoritmanın hızlı bir şekilde çalıştığı gözlemlenmiştir. Ancak, matris boyutları arttıkça, çözümün bulunma süresi artabilir. Bu durum, matris boyutlarındaki artışın çözüm olasılığını artırması ve ayrıca başlangıç durumunun rasgele seçilmesinin sonucu etkileyebileceği gerçeğinden kaynaklanmaktadır.

Bu algoritmanın hızının diğer internet üzerindeki benzer algoritmalarla karşılaştırıldığında belirgin bir şekilde daha hızlı olduğu görülmektedir. Ancak, unutulmamalıdır ki matris boyutları ve başlangıç durumu gibi faktörler, çözüm süresinde değişikliklere neden olabilir. Bu nedenle, programın performansını değerlendirirken bu faktörleri dikkate almak önemlidir.



## SONUÇ

Sonuç olarak, çeşitli arama algoritmalarını kullanarak çözülen dinamik ölçekli bir kaydırma bulmacasının performansını değerlendiren bu çalışma, her algoritmanın belirli avantajlara ve dezavantajlara sahip olduğunu gösteriyor. Geniş Öncelikli Arama, düşük maksimum depolama kullanımı ve işlenen düğüm sayısına rağmen, çözüm süresinde bir miktar uzunluğa sahiptir. Derinlik Öncelikli Arama ise işlenen düğüm ve maksimum depolama sayıları bakımından biraz daha yüksek değerlere sahip olmasına rağmen, çözüm süresi biraz daha kısa. Uniform Cost Search, çözüm süresi ve işlenen düğüm sayısı açısından orta seviyede bir performans sergilerken, Derinlik Sınırlı Arama, düşük çözüm süresi ve işlenen düğüm sayısı ile dikkat çekiyor. İteratif Derinlemesine Arama, genel olarak makul performans gösterirken, A\* Arama daha uzun çözüm süreleriyle birlikte daha fazla işlenen düğüm ve maksimum depolama kullanımına sahiptir. Açgözlü Arama ise düşük çözüm süresi ve makul düğüm işleme sayılarıyla etkileyici bir performans sergiler. Bu değerlendirme, belirli durum ve kısıtlamalara bağlı olarak hangi algoritmanın tercih edilebileceğine dair bir genel bakış sunmaktadır.