

» بے نام خدا «

درس : مباحث ویرہ

استاد : محمد احمد زارہ

رشتہ : مهندس کامپیوٹر

دانشجو : زمر اے بی امل - زمر اے امل

دانشگاہ : ملی تعلیمات اکوڑشہ میٹاب

Subject.

Day. Month. Year.

1409, 01, 20: Data structures and Algorithms

Array و List و Queue و Stack

A

در ریاضیات، آرایه (Array) و لیست (List) از انواع ساختارهای داده هستند که برای ذخیره و دسترسی به داده‌ها استفاده می‌شوند. اما تفاوت اصلی آنها در نحوه دسترسی به داده‌هاست.

آرایه (Array) یک ساختار داده است که از یک یا چند عنصر از یک نوع تشکیل شده است. هر عنصر در آرایه با یک آدرس مشخص (Index) قابل دسترسی است.

لیست (List) یک ساختار داده است که از یک یا چند عنصر از یک یا چند نوع تشکیل شده است. هر عنصر در لیست با یک آدرس مشخص (Index) قابل دسترسی است.

تفاوت اصلی بین آرایه و لیست در نحوه دسترسی به داده‌هاست. در آرایه، دسترسی به داده‌ها با استفاده از آدرس مشخص (Index) امکان‌پذیر است. در لیست، دسترسی به داده‌ها با استفاده از آدرس مشخص (Index) امکان‌پذیر نیست.

در آرایه، دسترسی به داده‌ها با استفاده از آدرس مشخص (Index) امکان‌پذیر است. در لیست، دسترسی به داده‌ها با استفاده از آدرس مشخص (Index) امکان‌پذیر نیست.

در آرایه، دسترسی به داده‌ها با استفاده از آدرس مشخص (Index) امکان‌پذیر است. در لیست، دسترسی به داده‌ها با استفاده از آدرس مشخص (Index) امکان‌پذیر نیست.

در آرایه، دسترسی به داده‌ها با استفاده از آدرس مشخص (Index) امکان‌پذیر است. در لیست، دسترسی به داده‌ها با استفاده از آدرس مشخص (Index) امکان‌پذیر نیست.

در آرایه، دسترسی به داده‌ها با استفاده از آدرس مشخص (Index) امکان‌پذیر است. در لیست، دسترسی به داده‌ها با استفاده از آدرس مشخص (Index) امکان‌پذیر نیست.

در آرایه، دسترسی به داده‌ها با استفاده از آدرس مشخص (Index) امکان‌پذیر است. در لیست، دسترسی به داده‌ها با استفاده از آدرس مشخص (Index) امکان‌پذیر نیست.

10

Day Month Year

11. Unbalanced B-Tree, Binary Tree

11. Blackberry Tree, Blackberry Tree - 6
 12. Blackberry Tree, Blackberry Tree - 6
 13. Blackberry Tree, Blackberry Tree - 6

مجلس شورای اسلامی
کتابخانه و اسناد
کتابخانه و اسناد

John Williams
John Williams
John Williams

1. The first case is $B = 0$.

1) $\frac{1}{2} \frac{d}{dt} \left(\frac{1}{2} m v^2 \right) = \frac{1}{2} m v \frac{dv}{dt}$

مولا به نونه ای طرای شیده به حلالی مکتوبی به درصیب یافوده است

15h 4-1

الحمد لله الذي هدانا لهذا
 ما كنا لنهتدي لولا أن هدانا الله

Curiosities of the World of the 19th Century

Handwritten note: ...

ماتنه درخت ها، انحصار انبري و درختان عال

AM و درخت هاى قزق - ليلى و مقبول

[illegible]

مجلسه علمیه در روز پنجشنبه ۱۳۰۲/۱۲/۱۲

[illegible]

K. A. B. Co. - 12

capitulum Bacc. Del., involucrum callos.

classmate to a T_{L_1} (0-68) and a

مجلس استشاری عالی در امور معارف و اوقاف و صنایع مستظرفه

مجلس الامم المتحدة
مجلس الامم المتحدة

موقوفه دایم، برای احداث و تکمیل و تعمیر و نگهداری و اجاره و فروش و سایر امور مربوطه

admirable design for a book of the
- 11 -

المجلس العلمي في دار الحديث

Subject

Day: _____ Month: _____ Year: _____

DYNAMIC PROGRAMMING I

Day: _____ Month: _____ Year: _____
 Dynamic Programming
 1. مسائل که در آن می توان به کمک جواب های قبلی مسئله را حل کرد
 2. DP مسائل بهینه سازی است که در آن می توان به کمک جواب های قبلی مسئله را حل کرد
 3. DP مسائل بهینه سازی است که در آن می توان به کمک جواب های قبلی مسئله را حل کرد
 4. DP مسائل بهینه سازی است که در آن می توان به کمک جواب های قبلی مسئله را حل کرد
 5. DP مسائل بهینه سازی است که در آن می توان به کمک جواب های قبلی مسئله را حل کرد
 6. DP مسائل بهینه سازی است که در آن می توان به کمک جواب های قبلی مسئله را حل کرد
 7. DP مسائل بهینه سازی است که در آن می توان به کمک جواب های قبلی مسئله را حل کرد
 8. DP مسائل بهینه سازی است که در آن می توان به کمک جواب های قبلی مسئله را حل کرد
 9. DP مسائل بهینه سازی است که در آن می توان به کمک جواب های قبلی مسئله را حل کرد
 10. DP مسائل بهینه سازی است که در آن می توان به کمک جواب های قبلی مسئله را حل کرد

[illegible]

کاربرد در مسائل مختلف: DP در انواع مسائل مانند مسیریابی و مسائل ترکیبیاتی و مسائل مربوط به زمان بندی و مسائل ماکزیمم کردن و ... این نوع کاربردها نشان دهنده قدرت و انعطاف پذیری DP است.

بیا بر این مسائل که در آن صورت بازگشت می نمودند و معین است
مطابق فایده داشته باشد و استفاده از ۵۲۲ زمان های چند جلی ای که در
درماند این امر و نیز در مسائل با افتاز و بودی خرد است و در

Subject

Day _____ Month _____ Year _____

Interlocking Cast Dash Table

[illegible]

مکتبہ اسلامیہ، لاہور

Handwritten notes in Persian script, likely a student's work or a teacher's notes, mentioning "Math Table" and "Array".

کرن (Array) : ہر عنصر کے لیے ایک نمبر ہوتا ہے۔

در وقت شام در کلاس Table و در وقت صبح در کلاس Table و در وقت ظهر در کلاس Table و در وقت عصر در کلاس Table و در وقت شب در کلاس Table و در وقت روز در کلاس Table و در وقت...

اگر کسی را حال بد شود تا آنکه کسی را بگوید که این را بخورد
برای دفعه و چنانچه در این روزها در حال بد شود

ملاحظات: در صورتی که اطلاعات حاصل از این روش، به گونه‌ای باشد که بتواند به عنوان یک معیار برای ارزیابی و مقایسه بین سیستم‌های مختلف استفاده شود.

[illegible]

مجلسی قضا علی بن عباس کی دو کتابوں مختلف نسخوں میں ہیں جن میں سے ایک نسخہ قضا علی بن عباس کی ہے

[illegible]

یاد این میوه نازک و شیرین و خوش طعم است . پخته و با شکر و دارچین : برای دیابت و کمبود قند
و کلسیم و برای معده و ریه ها . با آب و شکر و دارچین : برای دیابت و کمبود قند و کلسیم

[illegible]

سید احمد علی خان

Subject

Day Month Year

Recursion
به تعبیر ساده تر، است که در آن یک تابع خودش را می‌خواند تا به یک حالت پایه (Base Case) برسد. این حالت پایه، نقطه‌ای است که تابع دیگر تابع خودش را نخواند و فقط یک مقدار را برمی‌گرداند.

مثال: محاسبه فیبوناچی

در دنباله فیبوناچی، هر عدد از مجموع دو عدد قبلی به دست می‌آید. برای محاسبه عدد n ام، ما به دو عدد قبلی نیاز داریم. این می‌تواند به صورت بازگشتی تعریف شود:

فایبوناچی (Fibonacci):
در این حالت، n و $n-1$ به ترتیب به $n-2$ و $n-3$ می‌روند تا به حالت پایه برسیم.

نکته: در این مثال، $n=0$ و $n=1$ به عنوان حالت پایه در نظر گرفته می‌شوند. برای $n=0$ ، مقدار 0 و برای $n=1$ ، مقدار 1 برمی‌گردد.

در این مثال، n و $n-1$ به ترتیب به $n-2$ و $n-3$ می‌روند تا به حالت پایه برسیم.

نکته: در این مثال، n و $n-1$ به ترتیب به $n-2$ و $n-3$ می‌روند تا به حالت پایه برسیم.

Signature

Subject _____

Day _____ Month _____ Year _____

2. Indexing of List & Table

1. When a list or table is created, it is automatically indexed. The index is a list of the first few characters of each item in the list or table. This makes it easy to find a specific item.

2. To create an index for a list or table, use the `Index` property. For example, to create an index for a list named `myList`, you would use the following code:

```
myList.Index = 1
```

3. The `Index` property is a number that represents the starting point for the index. The default value is 1, which means that the index starts at the first item in the list or table.

4. You can also specify the number of items to include in the index. For example, to create an index for a list named `myList` that includes the first 10 items, you would use the following code:

```
myList.Index = 1, 10
```

5. The `Index` property is also used to access items in the index. For example, to access the first item in the index for a list named `myList`, you would use the following code:

```
myList.Index(1)
```

6. The `Index` property is also used to remove items from the index. For example, to remove the first item from the index for a list named `myList`, you would use the following code:

```
myList.Index(1) = Nothing
```

7. The `Index` property is also used to add items to the index. For example, to add a new item to the index for a list named `myList`, you would use the following code:

```
myList.Index(1) = "New Item"
```

8. The `Index` property is also used to sort the index. For example, to sort the index for a list named `myList` in ascending order, you would use the following code:

```
myList.Index.Sort()
```

9. The `Index` property is also used to filter the index. For example, to filter the index for a list named `myList` to only include items that start with the letter 'A', you would use the following code:

```
myList.Index.Where(Function(x) x.StartsWith("A"))
```

10. The `Index` property is also used to group the index. For example, to group the index for a list named `myList` by the first letter of each item, you would use the following code:

```
myList.Index.GroupBy(Function(x) x.FirstChar())
```

11. The `Index` property is also used to join the index. For example, to join the index for a list named `myList` into a single string, you would use the following code:

```
myList.Index.Join()
```

12. The `Index` property is also used to convert the index to a different data type. For example, to convert the index for a list named `myList` to a string array, you would use the following code:

```
myList.Index.ToArray()
```

13. The `Index` property is also used to convert the index to a different data type. For example, to convert the index for a list named `myList` to a list of strings, you would use the following code:

```
myList.Index.ToList()
```

14. The `Index` property is also used to convert the index to a different data type. For example, to convert the index for a list named `myList` to a dictionary, you would use the following code:

```
myList.Index.ToDictionary()
```

15. The `Index` property is also used to convert the index to a different data type. For example, to convert the index for a list named `myList` to a JSON string, you would use the following code:

```
myList.Index.ToJson()
```

16. The `Index` property is also used to convert the index to a different data type. For example, to convert the index for a list named `myList` to a CSV string, you would use the following code:

```
myList.Index.ToCsv()
```

17. The `Index` property is also used to convert the index to a different data type. For example, to convert the index for a list named `myList` to a XML string, you would use the following code:

```
myList.Index.ToXml()
```

18. The `Index` property is also used to convert the index to a different data type. For example, to convert the index for a list named `myList` to a YAML string, you would use the following code:

```
myList.Index.ToYaml()
```

19. The `Index` property is also used to convert the index to a different data type. For example, to convert the index for a list named `myList` to a BSON string, you would use the following code:

```
myList.Index.ToBson()
```

20. The `Index` property is also used to convert the index to a different data type. For example, to convert the index for a list named `myList` to a BSON document, you would use the following code:

```
myList.Index.ToBsonDocument()
```


Subject

Day Month Year

1. **Set** و **Python** چارای خاصیت ها برای استفاده از آن است.
2. **Python** و **Set** در پایتون یک مجموعه است که دارای ویژگی های زیر می باشد:
3. **غیر مرتبه ای**: در پایتون مجموعه ها به ترتیب خاصی ذخیره نمی شوند و به همین دلیل نمی توان به یک عضو از مجموعه با استفاده از اندیس دسترسی داشت.
4. **غیر تکراری**: در پایتون هر عضو فقط یک بار در مجموعه ظاهر می شود و تکرار آن بی فایده است.
5. **غیر متغیر**: در پایتون نمی توان به یک عضو از مجموعه دسترسی داشت و آن را تغییر داد.
6. **Python** `my_list = [1, 2, 3, 4, 5]` `unique_values = set(my_list)` `print(unique_values)` خروجی: `{1, 2, 3, 4, 5}`
7. **عملیات**: مجموعه ها دارای عملیات زیر می باشند:
8. **تفاوت**: `set1 - set2` تفاوت بین `set1` و `set2` را می دهد.
9. **اتصال**: `set1 | set2` اتصالی بین `set1` و `set2` را می دهد.
10. **تقاطع**: `set1 & set2` تقاطع بین `set1` و `set2` را می دهد.
11. **تفاوت متقابل**: `set1 ^ set2` تفاوت متقابل بین `set1` و `set2` را می دهد.
12. **تفاوت**: `set1.difference(set2)` تفاوت بین `set1` و `set2` را می دهد.
13. **تفاوت متقابل**: `set1.symmetric_difference(set2)` تفاوت متقابل بین `set1` و `set2` را می دهد.
14. **تفاوت**: `set1.difference_update(set2)` تفاوت بین `set1` و `set2` را می دهد.
15. **تفاوت**: `set1.intersection_update(set2)` تقاطع بین `set1` و `set2` را می دهد.
16. **تفاوت**: `set1.union_update(set2)` اتصالی بین `set1` و `set2` را می دهد.
17. **تفاوت**: `set1.update(set2)` اتصالی بین `set1` و `set2` را می دهد.
18. **تفاوت**: `set1.add(element)` عضو `element` را به `set1` اضافه می کند.
19. **تفاوت**: `set1.remove(element)` عضو `element` را از `set1` حذف می کند.
20. **تفاوت**: `set1.pop()` یک عضو را از `set1` حذف می کند.

Day. Month. Year.

Cyanocephalus stock, above E.

James Mack, Queen Elizabeth
independence was won population only
Elizabeth Mack Wilson

2. Chips (thinly cut) (1st cut)

Push
Up
Stack

Week 6 (FD)

[illegible]

Wetland - wetlands are areas that are periodically flooded or saturated with water.

George Washington College, Washington, D.C.
 Department of the History of the United States

Explain the role of the GPIIb/IIIa receptor in platelet aggregation.

2. I will help him and we can

Remonty Gibrikoche Capon de Lucerne 1897
 1900 11.11.18 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2658 2659 2660 2661 2662 2663 2664 2665 2666 2667 2668 2669 2670 2671 2672 2673 2674 2675 2676 2677 2678 2679 2680 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708 270