**Алгоритми – Jid-класификация** ( ориентировъчна ) 2016/2019/2020

# STL - контейнери и алгоритми

|  |  |
| --- | --- |
| * контейнери: vector, stack, queue, deque, priority\_queue, set, pair, map * алгоритми: sort(), reverse(), next\_permutation(), prev\_permutation(), lower\_bound(), upper\_bound() ..... | от D нагоре  от C нагоре |

# Теория на числата:

|  |  |
| --- | --- |
| * Делители. Прости числа – стандартен алгоритъм, решето на Ератостен * НОД, НОК – Евклид * Прости множители, **Факторизация** * РАЕ - същност, решаване на Диофантови уравнения | all  от D нагоре  от D нагоре  от С нагоре |

# Числови последователности. Едномерни масиви

|  |  |
| --- | --- |
| * търсене, броене * мин/макс, редици и подредици .... * двоично търсене .... * сортиране с настройка на библиотечна функция sort() * сливане (merge) на сортирани масиви | all  all  от D нагоре  от D нагоре  от D нагоре |

# Символи и Низове

|  |  |
| --- | --- |
| * обработка на символи и текст, проверки, преобразуване (цифра⬄число, малкаБ ⬄глБ, …) * оптимално търсене в низ (KMP, Aho-Corasick, Z-алгоритъм, … Хеширане …) * дълги числа – същност, операции (сравняване, сбор, разлика, произведение) * дълги числа – частно, остатък * Обратен Полски Запис (изчисляване, създаване) | all  от C нагоре  от D нагоре  от C нагоре  от C нагоре |

# Бройни системи

|  |  |
| --- | --- |
| * същност, видове * преобразуване – всички варианти (2-10, 10-2, P-10, 10-P, P-Q) | от D нагоре  от D нагоре |

# Комбинаторика. Множества

|  |  |
| --- | --- |
| * комбинаторни конфигурации – видове, принципи (на събирането, на умножеието) * генериране на комбинаторни конфигурации – най-вече лексикографска наредба * кодиране на пермутации * декодиране на пермутации * комбинаторни числа (Бел, Стирлинг, ….) * код на Грей * триъгълник на Паскал, Нютонов бином | от D нагоре  от D нагоре  от D нагоре  от D нагоре  от D нагоре  от D нагоре  от D нагоре |

* Разбиване на число като сума от естествени числа
* Разбиване на число на сума дадени числа (монети)

# Динамично оптимиране (ДО)

|  |  |
| --- | --- |
| * Сума в триъгълник, правоъгълник, път на североизто * Братска подялба * Брой разбивания на суми/монети * Раница * Най-дълга растяща подредица * Най-дълга обща подредица * Максимална сума в подправоъгълник * Метод на хистограмата * Спортни срещи * Задача за назначенинята | от D нагоре  от D нагоре  от D нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре |

# Геометрия

|  |  |
| --- | --- |
| * уравнение на права (общо, с ъглов коефициент) * ориентирана 3-ка точки, детерминанта, * проверка, дали 3 точки лежат на 1 права (дали образуват триъгълник) * разстояние от точка до права * проверка за пресичане на отсечки * проверка, дали точка е вътрешна за 3-ъгълник/многоъгълник * пресечна точка на отсечки (прави) * лице на триъгълник/многоъгълник ( с детерминанта, с формула за трапец) * подреждане на върховете на ногоъгълник * изпъкнала обвивка | от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре |

# АСД - Граф  - видове и алгоритми

|  |  |
| --- | --- |
| * приложения на **Dfs** (обхождане, проверка за цикъл, има ли път, всички пътища, брой свързани компоненти, **Ойлеров** граф - път/цикъл, ...) * приложения на **Dfs** **Bfs** (обхождане, минимален път в граф без тегла) * min път в граф с тегла   = **Dijkstra** (в граф с **+** тегла **–** от връх до всички останали) O(E + |V|.log|V|) по време, ако е с приоритетна опашка (с матрица на съседство е O (|V|2) и по време и по памет)  = **Belman**-**Ford** ( в граф с **+** и **–** тегла - от връх до всички останали) Ot (|V|3) Om(|V|.|E|)  = **Floyd** ( в граф с **+** и **–** тегла – min път между всички двойки върхове) O (|V|3)   * **MST** (Minimal Spininig Tree) **–** **Kruskal** или Prim * Потоци и мрежи **–** **MaxFlow**, двойкосъчетания * Силно свързани Компоненти * Двуделни графи * Артикулационни точки, Мостове (bridges) | от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от B нагоре  от B нагоре  от B нагоре |

# АСД – Дърво (двоично дърво) – използвани видове и алгоритми с тях

|  |  |
| --- | --- |
| * **BST** (Binary Search Tree) - обхождания * **SegmentTree**, Interval Tree * **IndexTree** (Fenwick) * **Treap** (Cartesian Tree) * **RMQ** (Range Minimal/Maximal Query) * **LCA** (Lowest Common Ancestor) | от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре  от C нагоре |

# Побитови операции - същност, видове, приложения, оптимизации с побитови операции

|  |  |
| --- | --- |
| * броене на 1-ци или 0-ли в 2-чното представяне на число * преобразуване (10)-(2) * маски | от C нагоре  от C нагоре  от C нагоре |

# Алгоритми и техники

|  |  |
| --- | --- |
| * Двоично търсене по отговора * Greedy алгоритми * бързо повдигане на степен * бързо умножение на матрици | от C нагоре  от C нагоре  от C нагоре  от B нагоре |

# Игри

|  |  |
| --- | --- |
| * Печеливши/губещи позиции * Четност/нечетност * Игри с пулове, карти * НИМ * Алфа-бета отсичане (за група А) | от C нагоре  от C нагоре  от C нагоре  от C нагоре  AB |

# Интерактивни задачи

(до скоро само за група А, но вече може да дават и за група В)

# Сортиране: Радикс, CountSort

28.02.2019, 02:49 ч.

14.08.2020, 18:08 ч.