

Een eerste probleem oplossen

UVa 11172 — Relational Operators

beOI Training



OLYMPIADE BELGE D'INFORMATIQUE
BELGISCHE INFORMATICA-OLYMPIADE

Inleiding

Dit document legt kort uit hoe je een omgeving opzet om in te programmeren en hoe je je eerste algoritmisch probleem op UVa oplost.

De uitleg concentreert zich op een Linux omgeving en het gebruik van de commandoregel. Deze instrumenten zijn krachtig en komen overeen met de beschikbare instrumenten tijdens een wedstrijd, dus is het aangeraden deze een beetje te kunnen gebruiken.

Om Linux te gebruiken zijn er meerdere opties, maar de meest eenvoudige oplossing om mee te beginnen is het installeren in een virtuele machine. Voor handleidingen, zoek naar “Ubuntu installeren op een virtuele machine”¹ op Google

¹Sommige instructies in dit document zijn specifiek voor Ubuntu, dus als je geen eigen mening hebt, kies dan best voor deze Linus distributie

Inhoudstafel

Vereisten

Programmeren

Compileren, testen, indienen

En verder

programmeer omgeving (C++)

Op Linux:

- ▶ Installeer de compiler g++: `sudo apt install g++`
- ▶ Gebruik een goede tekst editor: de standaard editor is perfect (gedit, geany, etc.). Meer geavanceerd: vim, emacs.














Op Windows:

- ▶ De beste oplossing: installeer Linux (dual boot of VM)
- ▶ Anders, bekijk Code::Blocks:
<http://www.codeblocks.org/downloads/binaries>
- ▶ Download het bestand dat lijkt op `codeblocks-16.01mingw-setup.exe`
- ▶ Gebruik **geen** online IDE! (zeer slechte gewoonte)

Online judge

- ▶ Online judge = systeem dat code evalueert en het gedrag ervan verifieert op veel voorbeelden
- ▶ We gebruiken UVA (maak een account aan):
<http://uva.onlinejudge.org>

Registration

Name:	<input type="text" value="Gennady Korotkevich"/>	 
Email:	<input type="text" value="gennady@korotkevi.ch"/>	 
Username:	<input type="text" value="tourist"/>	 
Password:	<input type="password" value="....."/>	 
Verify Password:	<input type="password" value="....."/>	 
Former UVA ID:	<input type="text"/>	
Results email:	<input type="checkbox"/>	
Virtual Judge:	<input type="checkbox"/>	



I'm not a robot



reCAPTCHA

[Privacy](#) - [Terms](#)

Inhoudstafel

Vereisten

Programmeren

Compileren, testen, indienen

En verder

Inleiding van het probleem

Probleemtekst:

<http://uva.onlinejudge.org/external/111/11172.pdf>

Some operators checks about the relationship between two values and these operators are called relational operators.

Given two numerical values your job is just to **find out the relationship between them** that is (i) First one is **greater than** the second (ii) First one is **less than** the second or (iii) First and second one is **equal**.

De context diagonaal lezen (een talent om te ontwikkelen).

Invoerformaat

De invoer wordt gegeven op *standaard invoer*, alsof iemand het bestand manueel lijn per lijn zou intypen in de console.

Input

First line of the input file is an integer t ($t < 15$) which denotes how many sets of inputs are there. Each of the next t lines contain two integers a and b ($|a|, |b| < 1000000001$).

Sample input

3

10 20

20 10

10 10

Input: test cases

De invoer bestaat vaak uit meerdere test cases die zich in hetzelfde bestand bevinden.

Hier is het aantal test cases t gegeven in het begin.

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int t; // aantal test cases
    cin >> t;
    for (int i=0; i<t; i++) {
        // code hier
    }
}
```

Input: gegevens

Nu volstaat het a en b in te lezen.

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int t; // aantal test cases
    cin >> t;
    for (int i=0; i<t; i++) {
        int a, b; // gegevens
        cin >> a >> b;
        // bereken het antwoord
    }
}
```

Opmerking: cin laat toe om gehele getallen, floating point getallen en string in te lezen, gescheiden door spaties of op een nieuwe regel.

Resultaat en uitvoer

Nu we a en b hebben, moeten we enkel het resultaat nog berekenen.

```
// in de lus:  
if (a < b)  
    cout << "<" << endl;  
else if (a > b)  
    cout << ">" << endl;  
else  
    cout << "=" << endl;
```

Opmerking: Het is geen probleem om uitvoer te beginnen schrijven voor je alle invoer gelezen hebt.

Inhoudstafel

Vereisten

Programmeren

Compileren, testen, indienen

En verder

De console openen

De uitleg voor de compilatie en het testen concentreren zich op een Linux omgeving in de console.

Twee manieren om de console te openen:

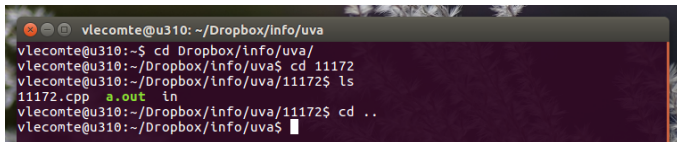
- ▶ Openen in de lijst van toepassingen. Typisch heet het “Terminal”, “Console” of “Shell”
- ▶ Vanaf de verkenner: naar de map navigeren, rechts klikken en “Open in Terminal” of gelijkaardig klikken. Voordeel: we zijn direct in de juiste map.

Wanneer je de console open, geeft een regel tekst de huidige map aan: standaard is dat ~, de map “Home”, die Documents, Downloads, ... bevat

Basiscommando's

Nuttige commando's:

- ▶ `ls`: de inhoud van de huidige map ophoofden
- ▶ `cd folder`: naar de submap folder gaan
- ▶ `cd ..`: teruggaan naar de bovenliggende map



```
vlecomte@u310: ~/Dropbox/info/uva
vlecomte@u310:~$ cd Dropbox/info/uva/
vlecomte@u310:~/Dropbox/info/uva$ cd 11172
vlecomte@u310:~/Dropbox/info/uva/11172$ ls
11172.cpp  a.out  in
vlecomte@u310:~/Dropbox/info/uva/11172$ cd ..
vlecomte@u310:~/Dropbox/info/uva$
```

Nuttige toetsen:

- ▶ Tab: de naam van een bestand/map/programma die gedeeltelijk getypt is aanvullen.
- ▶ Pijltje naar boven/beneden: de geschiedenis doorlopen en eerder uitgevoerde commando's herhalen.

Compileren

Commando om te compileren:

```
g++ -std=c++11 -Wall -Wextra -Wshadow -O2 a.cpp
```

- ▶ g++: naam van de compiler
- ▶ -std=c++11: gebruikte C++ versie
- ▶ -Wall -Wextra -Wshadow: activeer veel waarschuwingen om bugs te vermijden
- ▶ -O2: compileer het programma om het snel uit te voeren
- ▶ a.cpp: source code (je moet in de juiste map zijn!)

Als alles in orde is, geeft dit geen uitvoer en maakt het het programma a.out aan. Anders, lees de foutboodschap/waarschuwing, verbeter en probeer opnieuw.

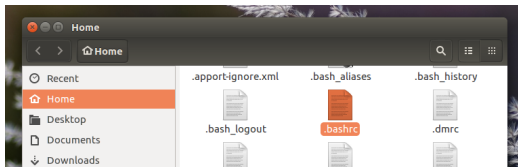
Om de naam van a.out aan te passen, voeg de optie -o andereNaam toe.

Compilatie: pro-tip

Raadgeving: voor de volgende regel toe op het einde van
~/.bashrc:

```
alias g="g++ -std=c++11 -Wall -Wextra -Wshadow -O2"
```

De naam `.bashrc` begint met een punt en is dus standaard verborgen. Om het weer te geven in de verkenner, gebruik `Ctrl+H`. Het is een gewoon tekstbestand.



Nadat je de console opnieuw geopend hebt, volstaat het op `g a.cpp` te typen om te compileren met alle opties.

Testen en debuggen

Om het programma uit te voeren, type je gewoon `./a.out`. Als je de invoer lijn per lijn geeft, zal het programma die inlezen en daarop reageren.²

Om te vermijden dat je de invoer elke keer opnieuw met intypen, kan je deze opslaan in een tekstbestand (bvb. `11172.in`, de naam heeft geen belang) en hem automatisch aan je programma geven met het commando `./a.out < 11172.in`. Voor grotere programma's is het aangeraden je eigen input te maken.

Gelijkaardig kan je de uitvoer van een programma in een tekstbestand opslaan met `./a.out > 11172.out` of zelfs de twee combineren met `./a.out < 11172.in > 11172.out`. Het uitvoerbestand wordt aangemaakt als het nog niet bestond.

²Sommige problemen vragen om de invoer tot “het einde van het bestand” (EOF) te lezen. Dat kan je aangeven met `Ctrl+D`.

Indienen en verdict

Na het testen, is het tijd om je programma in te dienen bij de judge om te verifiëren dat correct en snel genoeg is. Het eenvoudigst is de pagina “Quick Submit” in de menu op links in UVa. Voer het nummer van het probleem in, en kies C++11 als taal.

Daarna kan je het verdict vinden in “My submissions”:

- ▶ Accepted: het programma is correct, bravo!
- ▶ Wrong Answer: de uitvoer is fout
- ▶ Time Limit Exceeded: het programma is te traag
- ▶ Runtime Error: het programma is gecrasht
- ▶ Compilation Error: het programma compileert niet
- ▶ Presentation Error: het uitvoerformaat is incorrect

Inhoudstafel

Vereisten

Programmeren

Compileren, testen, indienen

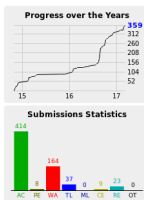
En verder

Selectieproblemen voor beCP

Nu je je eerste probleem opgelost hebt, kan je verdergaan met de 9 andere problemen voor de selectie voor stages van beCP. Meer info: <http://becp.be-oi.be/nl/>

Na ~ 1 , kan je ook uHunt gebruiken om je UVa statistieken te raadplegen: <http://uhunt.felix-halim.net>

Victor Lecomte (vlecomte) statistics



Last Submissions							Show : 5 10 50 100 500 ALL	
Problem		Verdict	Lang	Time	Best	Rank	Submit Time	
11172 - Relational Operator	discuss	Accepted	C++11	0.000	0.000	10422	3 days ago	
1062 - Containers	discuss	Accepted	C++11	0.000	0.000	494	6 days ago	
10935 - Throwing cards away I	discuss	Accepted	C++11	0.000	0.000	1799	6 days ago	
10935 - Throwing cards away I	discuss	Wrong answer	C++11	-	0.000	-	6 days ago	
11988 - Broken Keyboard (a...)	discuss	Accepted	C++11	0.210	0.000	2128	7 days ago	

Solved : 359, Submissions : 655

100 102 103 104 105 108 109 111 115 116 120 122 123 146 156 161 183 193 195 200 216 247 259 264 272 294 295 303 314 315 324 325 334 343 344 348 350 357 363 369 374 377 382 389 393 401 403 416 422 429 437 439 441 443 452 454 458 460 488 489 494 514 524 531 543 556 562 563 565 573 579 583 588 590 599 607 610 620 623 624 644 656 662 663 674 703 709 722 737 741 750 752 760 793 796 815 820 825 847 861 893 900 902 920 941 957 976 978 1047 1062 1064 1092 1096 1099 1111 1172 1174 1203 1211 1213 1218 1220 1222 1231 1232 1237 1244 1252 1254 1347 1572 1585 1656 1666 1669 1671 1678 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 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 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 2789 2790 2791 2792 2793 2794 2795 2796 2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818 2819 2820 2821 2822 2823 2824 2825 2826 2827 2828 2829 2830 2831 2832 2833 2834 2835 2836 2837 2838 2839 2840 2841 2842 2843 2844 2845 2846 2847 2848 2849 2850 2851 2852 2853 2854 2855 2856 2857 2858 2859 2860 2861 2862 2863 2864 2865 2866 2867 2868 2869 2870 2871 2872 2873 2874 2875 2876 2877 2878 2879 2880 2881 2882 2883 2884 2885 2886 2887 2888 2889 2890 2891 2892 2893 2894 2895 2896 2897 2898 2899 2900 2901 2902 2903 2904 2905 2906 2907 2908 2909 2910 2911 2912 2913 2914 2915 2916 2917 2918 2919 2920 2921 2922 2923 2924 2925 2926 2927 2928 2929 2930 2931 2932 2933 2934 2935 2936 2937 2938 2939 2940 2941 2942 2943 2944 2945 2946 2947 2948 2949 2950 2951 2952 2953 2954 2955 2956 2957 2958 2959 2960 2961 2962 2963 2964 2965 2966 2967 2968 2969 2970 2971 2972 2973 2974 2975 2976 2977 2978 2979 2980 2981 2982 2983 2984 2985 2986 2987 2988 2989 2990 2991 2992 2993 2994 2995 2996 2997 2998 2999 3000 3001 3002 3003 3004 3005 3006 3007 3008 3009 3010 3011 3012 3013 3014 3015 3016 3017 3018 3019 3020 3021 3022 3023 3024 3025 3026 3027 3028 3029 3030 3031 3032 3033 3034 3035 3036 3037 3038 3039 3040 3041 3042 3043 3044 3045 3046 3047 3048 3049 3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3066 3067 3068 3069 3070 3071 3072 3073 3074 3075 3076 3077 3078 3079 3080 3081 3082 3083 3084 3085 3086 3087 3088 3089 3090 3091 3092 3093 3094 3095 3096 3097 3098 3099 3100 3101 3102 3103 3104 3105 3106 3107 3108 3109 3110 3111 3112 3113 3114 3115 3116 3117 3118 3119 3120 3121 3122 3123 3124 3125 3126 3127 3128 3129 3130 3131 3132 3133 3134 3135 3136 3137 3138 3139 3140 3141 3142 3143 3144 3145 3146 3147 3148 3149 3150 3151 3152 3153 3154 3155 3156 3157 3158 3159 3160 3161 3162 3163 3164 3165 3166 3167 3168 3169 3170 3171 3172 3173 3174 3175 3176 3177 3178 3179 3180 3181 3182 3183 3184 3185 3186 3187 3188 3189 3190 3191 3192 3193 3194 3195 3196 3197 3198 3199 3200 3201 3202 3203 3204 3205 3206 3207 3208 3209 3210 3211 3212 3213 3214 3215 3216 3217 3218 3219 3220 3221 3222 3223 3224 3225 3226 3227 3228 3229 3230 3231 3232 3233 3234 3235 3236 3237 3238 3239 3240 3241 3242 3243 3244 3245 3246 3247 3248 3249 3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263 3264 3265 3266 3267 3268 3269 3270 3271 3272 3273 3274 3275 3276 3277 3278 3279 3280 3281 3282 3283 3284 3285 3286 3287 3288 3289 3290 3291 3292 3293 3294 3295 3296 3297 3298 3299 3300 3301 3302 3303 3304 3305 3306 3307 3308 3309 3310 3311 3312 3313 3314 3315 3316 3317 3318 3319 3320 3321 3322 3323 3324 3325 3326 3327 3328 3329 3330 3331 3332 3333 3334 3335 3336 3337 3338 3339 3340 3341 3342 3343 3344 3345 3346 3347 3348 3349 3350 3351 3352 3353 3354 3355 3356 3357 3358 3359 3360 3361 3362 3363 3364 3365 3366 3367 3368 3369 3370 3371 3372 3373 3374 3375 3376 3377 3378 3379 3380 3381 3382 3383 3384 3385 3386 3387 3388 3389 3390 3391 3392 3393 3394 3395 3396 3397 3398 3399 3400 3401 3402 3403 3404 3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3418 3419 3420 3421 3422 3423 3424 3425 3426 3427 3428 3429 3430 3431 3432 3433 3434 3435 3436 3437 3438 3439 3440 3441 3442 3443 3444 3445 3446 3447 3448 3449 3450 3451 3452 3453 3454 3455 3456 3457 3458 3459 3460 3461 3462 3463 3464 3465 3466 3467 3468 3469 3470 3471 3472 3473 3474 3475 3476 3477 3478 3479 3480 3481 3482 3483 3484 3485 3486 3487 3488 3489 3490 3491 3492 3493 3494 3495 3496 3497 3498 3499 3500 3501 3502 3503 3504 3505 3506 3507 3508 3509 3510 3511 3512 3513 3514 3515 3516 3517 3518 3519 3520 3521 3522 3523 3524 3525 3526 3527 3528 3529 3530 3531 3532 3533 3534 3535 3536 3537 3538 3539 3540 3541 3542 3543 3544 3545 3546 3547 3548 3549 3550 3551 3552 3553 3554 3555 3556 3557 3558 3559 3560 3561 3562 3563 3564 3565 3566 3567 3568 3569 3570 3571 3572 3573 3574 3575 3576 3577 3578 3579 3580 3581 3582 3583 3584 3585 3586 3587 3588 3589 3590 3591 3592 3593 3594 3595 3596 3597 3598 3599 3600 3601 3602 3603 3604 3605 3606 3607 3608 3609 3610 3611 3612 3613 3614 3615 3616 3617 3618 3619 3620 3621 3622 3623 3624 3625 3626 3627 3628 3629 3630 3631 3632 3633 3634 3635 3636 3637 3638 3639 3640 3641 3642 3643 3644 3645 3646 3647 3648 3649 3650 3651 3652 3653 3654 3655 3656 3657 3658 3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3674 3675 3676 3677 3678 3679 3680 3681 3682 3683 3684 3685 3686 3687 3688 3689 3690 3691 3692 3693 3694 3695 3696 3697 3698 3699 3700 3701 3702 3703 3704 3705 3706 3707 3708 3709 3710 3711 3712 3713 3714 3715 3716 3717 3718 3719 3720 3721 3722 3723 3724 3725 3726 3727 3728 3729 3730 3731 3732 3733 3734 3735 3736 3737 3738 3739 3740 3741 3742 3743 3744 3745 3746 3747 3748 3749 3750 3751 3752 3753 3754 3755 3756 3757 3758 3759 3760 3761 3762 3763 3764 3765 3766 3767 3768 3769 3770

Trainingsmaterialen

Alle materialen van de beCP training zijn publiek beschikbaar op de site: <https://github.com/be-oi/beoi-training>

- ▶ Per hoofdstuk geschikt, met duidelijk opschreven onderwerpen
- ▶ Slides van alle lessen
- ▶ Oefeningen verbonden aan de geziene onderwerpen

Andere sites stellen een nogal compleet programma van lessen en oefeningen voor:

- ▶ [France-IOI](#) (Frans): trainingssite van het Franse team voor IOI.
- ▶ [USACO](#) (Engels): idem voor de VS.

Platformen en wedstrijden

Er bestaan nog veel andere sites met oefeningen en wedstrijden:

- ▶ **Codeforces**: goede problemen en regelmatig wedstrijden
- ▶ **Kattis**: kwaliteitsproblemen, met aanduiding van niveau³
- ▶ **COCI**: 7 wedstrijden per jaar, erg gericht op IOI
- ▶ **USACO**: 4 wedstrijden per jaar, erg gericht op IOI, deelnemen eender wanneer tijdens een weekend
- ▶ **CodeChef**: drie wedstrijden per maand, waarvan 1 verspreid over tien dagen
- ▶ **ACM-ICPC Live Archive**: problemen van ACM-ICPC, een universitaire wedstrijd algoritmiek
- ▶ **SPOJ**: Enorme bibliotheek aan problemen

³Soms moet je dit met een korrel zout nemen als het nog niet door veel mensen opgelost is.

Boeken rond algoritmiek

Enkele goede referenties voor competitive programming en algoritmiek in het algemeen:

- ▶ **Competitive Programmer's Handbook** (gratis): Volledig nieuw, maar ziet er goed geschreven en tamelijk volledig uit.
- ▶ **Competitive Programming 3** (betalend): Heel volledig, maar soms moeilijk om te lezen. Bevat problemen voor elk onderwerp, van variabele kwaliteit.
- ▶ **Algorithms (Sedgewick, Wayne)** (betalend): Een goede referentie rond algoritmiek in het algemeen.