

## Today's Content

1. First missing positive Integer

Q: Given  $arr(N)$ , find first missing positive number / Natural Numbers.

Note: first the number not in  $arr()$

$arr() = \{3, -2, 1, 2, 7\}$  :  $ans = 4$

$arr() = \{-9, 2, 6, 4, -8, 1, 3\}$   $ans = 5$

$arr() = \{1, 2, 5, 6, 4, 3\}$   $ans = 7$

$arr() = \{4, 2, 1, 3\}$   $ans = 5$

$arr() = \{-4, 8, 3, -1, 0\}$   $ans = 1$

$arr() = \{-8, -3, -1, -5\}$   $ans = 1$

Idea:  $arr(N) = \{a_0, a_1, a_2, \dots, a_{N-1}\}$

	If Not present	Present
Search 1 : * return 1		✓
Search 2 : * return 2		✓
Search 3 : * return 3		✓
⋮		
Search N : * return N		✓

# Note: If all elements from  $1..N$  are present  
1<sup>st</sup> missing number is  $N+1$

```
#for: for(int i=1; i<=N; i++) {  
    #iterate on arr() & search i  
    if i is not present: return i;  
    else goto next element  
}  
return N+1;
```

Tc:  $O(N \times N) = O(N^2)$  sc:  $O(1)$

Idea2: Sort arr()

```
for(int i=1; i<=N; i++) {  
    # Apply Binary Search on arr() for i  
    if i is not present: return ij  
} else goto next element
```

return N+1;

Tc:  $O(N \log N + N \log N) = O(N \log N)$  Sc:  $O(1)$

↳ Sort  
↳ BS, N Times

Idea3: Using sum of +ve elements

arr() = {1 2 3} # Both have same sum

arr() = {2 2 2}

Idea4: Insert all arr() elements in hashset hs

```
for(int i=1; i<=N; i++) {  
    # Search i in hashset hs.  
    if i is not present: return ij  
} else goto next element
```

return N+1;

Tc:  $O(N + N \cdot 1) = O(N)$  Sc:  $O(N)$  → Because hashset

# Idea5:

Hint1: We are only searching 1..N, other numbers are irrelevant.

Hint2:

arr() = {4 3 1 2} → {1 2 3 4}

ind i = i+1 else

else n = n-1 ind

arr[8] = {  
 0: 4  
 1: 2  
 2: 7  
 3: 6  
 4: 9  
 5: 1  
 6: 8  
 7: 3  
 8: 5  
 9: 6  
 10: 7  
 11: 8  
 12: 1  
 13: 3  
 14: 4  
 15: 6  
 16: 7  
 17: 8  
 18: 1  
 19: 3  
 20: 4  
 21: 6  
 22: 7  
 23: 8  
 24: 1  
 25: 3  
 26: 4  
 27: 6  
 28: 7  
 29: 8  
 30: 1  
 31: 3  
 32: 4  
 33: 6  
 34: 7  
 35: 8  
 36: 1  
 37: 3  
 38: 4  
 39: 6  
 40: 7  
 41: 8  
 42: 1  
 43: 3  
 44: 4  
 45: 6  
 46: 7  
 47: 8  
 48: 1  
 49: 3  
 50: 4  
 51: 6  
 52: 7  
 53: 8  
 54: 1  
 55: 3  
 56: 4  
 57: 6  
 58: 7  
 59: 8  
 60: 1  
 61: 3  
 62: 4  
 63: 6  
 64: 7  
 65: 8  
 66: 1  
 67: 3  
 68: 4  
 69: 6  
 70: 7  
 71: 8  
 72: 1  
 73: 3  
 74: 4  
 75: 6  
 76: 7  
 77: 8  
 78: 1  
 79: 3  
 80: 4  
 81: 6  
 82: 7  
 83: 8  
 84: 1  
 85: 3  
 86: 4  
 87: 6  
 88: 7  
 89: 8  
 90: 1  
 91: 3  
 92: 4  
 93: 6  
 94: 7  
 95: 8  
 96: 1  
 97: 3  
 98: 4  
 99: 6  
 100: 7  
 101: 8  
 102: 1  
 103: 3  
 104: 4  
 105: 6  
 106: 7  
 107: 8  
 108: 1  
 109: 3  
 110: 4  
 111: 6  
 112: 7  
 113: 8  
 114: 1  
 115: 3  
 116: 4  
 117: 6  
 118: 7  
 119: 8  
 120: 1  
 121: 3  
 122: 4  
 123: 6  
 124: 7  
 125: 8  
 126: 1  
 127: 3  
 128: 4  
 129: 6  
 130: 7  
 131: 8  
 132: 1  
 133: 3  
 134: 4  
 135: 6  
 136: 7  
 137: 8  
 138: 1  
 139: 3  
 140: 4  
 141: 6  
 142: 7  
 143: 8  
 144: 1  
 145: 3  
 146: 4  
 147: 6  
 148: 7  
 149: 8  
 150: 1  
 151: 3  
 152: 4  
 153: 6  
 154: 7  
 155: 8  
 156: 1  
 157: 3  
 158: 4  
 159: 6  
 160: 7  
 161: 8  
 162: 1  
 163: 3  
 164: 4  
 165: 6  
 166: 7  
 167: 8  
 168: 1  
 169: 3  
 170: 4  
 171: 6  
 172: 7  
 173: 8  
 174: 1  
 175: 3  
 176: 4  
 177: 6  
 178: 7  
 179: 8  
 180: 1  
 181: 3  
 182: 4  
 183: 6  
 184: 7  
 185: 8  
 186: 1  
 187: 3  
 188: 4  
 189: 6  
 190: 7  
 191: 8  
 192: 1  
 193: 3  
 194: 4  
 195: 6  
 196: 7  
 197: 8  
 198: 1  
 199: 3  
 200: 4  
 201: 6  
 202: 7  
 203: 8  
 204: 1  
 205: 3  
 206: 4  
 207: 6  
 208: 7  
 209: 8  
 210: 1  
 211: 3  
 212: 4  
 213: 6  
 214: 7  
 215: 8  
 216: 1  
 217: 3  
 218: 4  
 219: 6  
 220: 7  
 221: 8  
 222: 1  
 223: 3  
 224: 4  
 225: 6  
 226: 7  
 227: 8  
 228: 1  
 229: 3  
 230: 4  
 231: 6  
 232: 7  
 233: 8  
 234: 1  
 235: 3  
 236: 4  
 237: 6  
 238: 7  
 239: 8  
 240: 1  
 241: 3  
 242: 4  
 243: 6  
 244: 7  
 245: 8  
 246: 1  
 247: 3  
 248: 4  
 249: 6  
 250: 7  
 251: 8  
 252: 1  
 253: 3  
 254: 4  
 255: 6  
 256: 7  
 257: 8  
 258: 1  
 259: 3  
 260: 4  
 261: 6  
 262: 7  
 263: 8  
 264: 1  
 265: 3  
 266: 4  
 267: 6  
 268: 7  
 269: 8  
 270: 1  
 271: 3  
 272: 4  
 273: 6  
 274: 7  
 275: 8  
 276: 1  
 277: 3  
 278: 4  
 279: 6  
 280: 7  
 281: 8  
 282: 1  
 283: 3  
 284: 4  
 285: 6  
 286: 7  
 287: 8  
 288: 1  
 289: 3  
 290: 4  
 291: 6  
 292: 7  
 293: 8  
 294: 1  
 295: 3  
 296: 4  
 297: 6  
 298: 7  
 299: 8  
 300: 1  
 301: 3  
 302: 4  
 303: 6  
 304: 7  
 305: 8  
 306: 1  
 307: 3  
 308: 4  
 309: 6  
 310: 7  
 311: 8  
 312: 1  
 313: 3  
 314: 4  
 315: 6  
 316: 7  
 317: 8  
 318: 1  
 319: 3  
 320: 4  
 321: 6  
 322: 7  
 323: 8  
 324: 1  
 325: 3  
 326: 4  
 327: 6  
 328: 7  
 329: 8  
 330: 1  
 331: 3  
 332: 4  
 333: 6  
 334: 7  
 335: 8  
 336: 1  
 337: 3  
 338: 4  
 339: 6  
 340: 7  
 341: 8  
 342: 1  
 343: 3  
 344: 4  
 345: 6  
 346: 7  
 347: 8  
 348: 1  
 349: 3  
 350: 4  
 351: 6  
 352: 7  
 353: 8  
 354: 1  
 355: 3  
 356: 4  
 357: 6  
 358: 7  
 359: 8  
 360: 1  
 361: 3  
 362: 4  
 363: 6  
 364: 7  
 365: 8  
 366: 1  
 367: 3  
 368: 4  
 369: 6  
 370: 7  
 371: 8  
 372: 1  
 373: 3  
 374: 4  
 375: 6  
 376: 7  
 377: 8  
 378: 1  
 379: 3  
 380: 4  
 381: 6  
 382: 7  
 383: 8  
 384: 1  
 385: 3  
 386: 4  
 387: 6  
 388: 7  
 389: 8  
 390: 1  
 391: 3  
 392: 4  
 393: 6  
 394: 7  
 395: 8  
 396: 1  
 397: 3  
 398: 4  
 399: 6  
 400: 7  
 401: 8  
 402: 1  
 403: 3  
 404: 4  
 405: 6  
 406: 7  
 407: 8  
 408: 1  
 409: 3  
 410: 4  
 411: 6  
 412: 7  
 413: 8  
 414: 1  
 415: 3  
 416: 4  
 417: 6  
 418: 7  
 419: 8  
 420: 1  
 421: 3  
 422: 4  
 423: 6  
 424: 7  
 425: 8  
 426: 1  
 427: 3  
 428: 4  
 429: 6  
 430: 7  
 431: 8  
 432: 1  
 433: 3  
 434: 4  
 435: 6  
 436: 7  
 437: 8  
 438: 1  
 439: 3  
 440: 4  
 441: 6  
 442: 7  
 443: 8  
 444: 1  
 445: 3  
 446: 4  
 447: 6  
 448: 7  
 449: 8  
 450: 1  
 451: 3  
 452: 4  
 453: 6  
 454: 7  
 455: 8  
 456: 1  
 457: 3  
 458: 4  
 459: 6  
 460: 7  
 461: 8  
 462: 1  
 463: 3  
 464: 4  
 465: 6  
 466: 7  
 467: 8  
 468: 1  
 469: 3  
 470: 4  
 471: 6  
 472: 7  
 473: 8  
 474: 1  
 475: 3  
 476: 4  
 477: 6  
 478: 7  
 479: 8  
 480: 1  
 481: 3  
 482: 4  
 483: 6  
 484: 7  
 485: 8  
 486: 1  
 487: 3  
 488: 4  
 489: 6  
 490: 7  
 491: 8  
 492: 1  
 493: 3  
 494: 4  
 495: 6  
 496: 7  
 497: 8  
 498: 1  
 499: 3  
 500: 4  
 501: 6  
 502: 7  
 503: 8  
 504: 1  
 505: 3  
 506: 4  
 507: 6  
 508: 7  
 509: 8  
 510: 1  
 511: 3  
 512: 4  
 513: 6  
 514: 7  
 515: 8  
 516: 1  
 517: 3  
 518: 4  
 519: 6  
 520: 7  
 521: 8  
 522: 1  
 523: 3  
 524: 4  
 525: 6  
 526: 7  
 527: 8  
 528: 1  
 529: 3  
 530: 4  
 531: 6  
 532: 7  
 533: 8  
 534: 1  
 535: 3  
 536: 4  
 537: 6  
 538: 7  
 539: 8  
 540: 1  
 541: 3  
 542: 4  
 543: 6  
 544: 7  
 545: 8  
 546: 1  
 547: 3  
 548: 4  
 549: 6  
 550: 7  
 551: 8  
 552: 1  
 553: 3  
 554: 4  
 555: 6  
 556: 7  
 557: 8  
 558: 1  
 559: 3  
 560: 4  
 561: 6  
 562: 7  
 563: 8  
 564: 1  
 565: 3  
 566: 4  
 567: 6  
 568: 7  
 569: 8  
 570: 1  
 571: 3  
 572: 4  
 573: 6  
 574: 7  
 575: 8  
 576: 1  
 577: 3  
 578: 4  
 579: 6  
 580: 7  
 581: 8  
 582: 1  
 583: 3  
 584: 4  
 585: 6  
 586: 7  
 587: 8  
 588: 1  
 589: 3  
 590: 4  
 591: 6  
 592: 7  
 593: 8  
 594: 1  
 595: 3  
 596: 4  
 597: 6  
 598: 7  
 599: 8  
 600: 1  
 601: 3  
 602: 4  
 603: 6  
 604: 7  
 605: 8  
 606: 1  
 607: 3  
 608: 4  
 609: 6  
 610: 7  
 611: 8  
 612: 1  
 613: 3  
 614: 4  
 615: 6  
 616: 7  
 617: 8  
 618: 1  
 619: 3  
 620: 4  
 621: 6  
 622: 7  
 623: 8  
 624: 1  
 625: 3  
 626: 4  
 627: 6  
 628: 7  
 629: 8  
 630: 1  
 631: 3  
 632: 4  
 633: 6  
 634: 7  
 635: 8  
 636: 1  
 637: 3  
 638: 4  
 639: 6  
 640: 7  
 641: 8  
 642: 1  
 643: 3  
 644: 4  
 645: 6  
 646: 7  
 647: 8  
 648: 1  
 649: 3  
 650: 4  
 651: 6  
 652: 7  
 653: 8  
 654: 1  
 655: 3  
 656: 4  
 657: 6  
 658: 7  
 659: 8  
 660: 1  
 661: 3  
 662: 4  
 663: 6  
 664: 7  
 665: 8  
 666: 1  
 667: 3  
 668: 4  
 669: 6  
 670: 7  
 671: 8  
 672: 1  
 673: 3  
 674: 4  
 675: 6  
 676: 7  
 677: 8  
 678: 1  
 679: 3  
 680: 4  
 681: 6  
 682: 7  
 683: 8  
 684: 1  
 685: 3  
 686: 4  
 687: 6  
 688: 7  
 689: 8  
 690: 1  
 691: 3  
 692: 4  
 693: 6  
 694: 7  
 695: 8  
 696: 1  
 697: 3  
 698: 4  
 699: 6  
 700: 7  
 701: 8  
 702: 1  
 703: 3  
 704: 4  
 705: 6  
 706: 7  
 707: 8  
 708: 1  
 709: 3  
 710: 4  
 711: 6  
 712: 7  
 713: 8  
 714: 1  
 715: 3  
 716: 4  
 717: 6  
 718: 7  
 719: 8  
 720: 1  
 721: 3  
 722: 4  
 723: 6  
 724: 7  
 725: 8  
 726: 1  
 727: 3  
 728: 4  
 729: 6  
 730: 7  
 731: 8  
 732: 1  
 733: 3  
 734: 4  
 735: 6  
 736: 7  
 737: 8  
 738: 1  
 739: 3  
 740: 4  
 741: 6  
 742: 7  
 743: 8  
 744: 1  
 745: 3  
 746: 4  
 747: 6  
 748: 7  
 749: 8  
 750: 1  
 751: 3  
 752: 4  
 753: 6  
 754: 7  
 755: 8  
 756: 1  
 757: 3  
 758: 4  
 759: 6  
 760: 7  
 761: 8  
 762: 1  
 763: 3  
 764: 4  
 765: 6  
 766: 7  
 767: 8  
 768: 1  
 769: 3  
 770: 4  
 771: 6  
 772: 7  
 773: 8  
 774: 1  
 775: 3  
 776: 4  
 777: 6  
 778: 7  
 779: 8  
 780: 1  
 781: 3  
 782: 4  
 783: 6  
 784: 7  
 785: 8  
 786: 1  
 787: 3  
 788: 4  
 789: 6  
 790: 7  
 791: 8  
 792: 1  
 793: 3  
 794: 4  
 795: 6  
 796: 7  
 797: 8  
 798: 1  
 799: 3  
 800: 4  
 801: 6  
 802: 7  
 803: 8  
 804: 1  
 805: 3  
 806: 4  
 807: 6  
 808: 7  
 809: 8  
 810: 1  
 811: 3  
 812: 4  
 813: 6  
 814: 7  
 815: 8  
 816: 1  
 817: 3  
 818: 4  
 819: 6  
 820: 7  
 821: 8  
 822: 1  
 823: 3  
 824: 4  
 825: 6  
 826: 7  
 827: 8  
 828: 1  
 829: 3  
 830: 4  
 831: 6  
 832: 7  
 833: 8  
 834: 1  
 835: 3  
 836: 4  
 837: 6  
 838: 7  
 839: 8  
 840: 1  
 841: 3  
 842: 4  
 843: 6  
 844: 7  
 845: 8  
 846: 1  
 847: 3  
 848: 4  
 849: 6  
 850: 7  
 851: 8  
 852: 1  
 853: 3  
 854: 4  
 855: 6  
 856: 7  
 857: 8  
 858: 1  
 859: 3  
 860: 4  
 861: 6  
 862: 7  
 863: 8  
 864: 1  
 865: 3  
 866: 4  
 867: 6  
 868: 7  
 869: 8  
 870: 1  
 871: 3  
 872: 4  
 873: 6  
 874: 7  
 875: 8  
 876: 1  
 877: 3  
 878: 4  
 879: 6  
 880: 7  
 881: 8  
 882: 1  
 883: 3  
 884: 4  
 885: 6  
 886: 7  
 887: 8  
 888: 1  
 889: 3  
 890: 4  
 891: 6  
 892: 7  
 893: 8  
 894: 1  
 895: 3  
 896: 4  
 897: 6  
 898: 7  
 899: 8  
 900: 1  
 901: 3  
 902: 4  
 903: 6  
 904: 7  
 905: 8  
 906: 1  
 907: 3  
 908: 4  
 909: 6  
 910: 7  
 911: 8  
 912: 1  
 913: 3  
 914: 4  
 915: 6  
 916: 7  
 917: 8  
 918: 1  
 919: 3  
 920: 4  
 921: 6  
 922: 7  
 923: 8  
 924: 1  
 925: 3  
 926: 4  
 927: 6  
 928: 7  
 929: 8  
 930: 1  
 931: 3  
 932: 4  
 933: 6  
 934: 7  
 935: 8  
 936: 1  
 937: 3  
 938: 4  
 939: 6  
 940: 7  
 941: 8  
 942: 1  
 943: 3  
 944: 4  
 945: 6  
 946: 7  
 947: 8  
 948: 1  
 949: 3  
 950: 4  
 951: 6  
 952: 7  
 953: 8  
 954: 1  
 955: 3  
 956: 4  
 957: 6  
 958: 7  
 959: 8  
 960: 1  
 961: 3  
 962: 4  
 963: 6  
 964: 7  
 965: 8  
 966: 1  
 967: 3  
 968: 4  
 969: 6  
 970: 7  
 971: 8  
 972: 1  
 973: 3  
 974: 4  
 975: 6  
 976: 7  
 977: 8  
 978: 1  
 979: 3  
 980: 4  
 981: 6  
 982: 7  
 983: 8  
 984: 1  
 985: 3  
 986: 4  
 987: 6  
 988: 7  
 989: 8  
 990: 1  
 991: 3  
 992: 4  
 993: 6  
 994: 7  
 995: 8  
 996: 1  
 997: 3  
 998: 4  
 999: 6  
 1000: 7  
 1001: 8  
 1002: 1  
 1003: 3  
 1004: 4  
 1005: 6  
 1006: 7  
 1007: 8  
 1008: 1  
 1009: 3  
 1010: 4  
 1011: 6  
 1012: 7  
 1013: 8  
 1014: 1  
 1015: 3  
 1016: 4  
 1017: 6  
 1018: 7  
 1019: 8  
 1020: 1  
 1021: 3  
 1022: 4  
 1023: 6  
 1024: 7  
 1025: 8  
 1026: 1  
 1027: 3  
 1028: 4  
 1029: 6  
 1030: 7  
 1031: 8  
 1032: 1  
 1033: 3  
 1034: 4  
 1035: 6  
 1036: 7  
 1037: 8  
 1038: 1  
 1039: 3  
 1040: 4  
 1041: 6  
 1042: 7  
 1043: 8  
 1044: 1  
 1045: 3  
 1046: 4  
 1047: 6  
 1048: 7  
 1049: 8  
 1050: 1  
 1051: 3  
 1052: 4  
 1053: 6  
 1054: 7  
 1055: 8  
 1056: 1  
 1057: 3  
 1058: 4  
 1059: 6  
 1060: 7  
 1061: 8  
 1062: 1  
 1063: 3  
 1064: 4  
 1065: 6  
 1066: 7  
 1067: 8  
 1068: 1  
 1069: 3  
 1070: 4  
 1071: 6  
 1072: 7  
 1073: 8  
 1074: 1  
 1075: 3  
 1076: 4  
 1077: 6  
 1078: 7  
 1079: 8  
 1080: 1  
 1081: 3  
 1082: 4  
 1083: 6  
 1084: 7  
 1085: 8  
 1086: 1  
 1087: 3  
 1088: 4  
 1089: 6  
 1090: 7  
 1091: 8  
 1092: 1  
 1093: 3  
 1094: 4  
 1095: 6  
 1096: 7  
 1097: 8  
 1098: 1  
 1099: 3  
 1100: 4  
 1101: 6  
 1102: 7  
 1103: 8  
 1104: 1  
 1105: 3  
 1106: 4  
 1107: 6  
 1108: 7  
 1109: 8  
 1110: 1  
 1111: 3  
 1112: 4  
 1113: 6  
 1114: 7  
 1115: 8  
 1116: 1  
 1117: 3  
 1118: 4  
 1119: 6  
 1120: 7  
 1121

arr[] = {

0	1	2	3	4	5	6	7	8	9
5	-14	6	7	9	-10	2	3	1	12
9	2	-10	2	5	6	7	-10	9	
1		3	-14						

ind i = i+1 else  
else n = n-1 ind

i	Correct data	Take data to correct position
0	arr[0] = 5 arr[0] = 9 arr[0] = 1 # goto next	arr[0] = 5 $\Rightarrow$ arr[4] #swap arr[0] & arr[4] arr[0] = 9 $\Rightarrow$ arr[8] #swap arr[0] & arr[8] <i>Out of bounds / Invalid data / goto next</i>
1	arr[1] = -14	arr[1] = -14 $\Rightarrow$ arr[-15] <i>Out of bounds / Invalid data / goto next</i>
2	arr[2] = 6 arr[2] = -10	arr[2] = 6 $\Rightarrow$ arr[5] #swap arr[2] & arr[5] arr[2] = -10 $\Rightarrow$ arr[-11]; <i>Out of bounds / Invalid data / goto next</i>
3	arr[3] = 7 arr[3] = 2 arr[3] = -14	arr[3] = 7 $\Rightarrow$ arr[6] #swap arr[3] & arr[6] arr[3] = 2 $\Rightarrow$ arr[1] #swap arr[3] & arr[1] arr[3] = -14 $\Rightarrow$ arr[-15] <i>Out of bounds / Invalid data / goto next</i>
4	arr[4] = 5 # goto next	
5	arr[5] = 6 # goto next	
6	arr[6] = 7 # goto next	
7	arr[7] = 3 arr[7] = -10	arr[7] = 3 $\Rightarrow$ arr[2] #swap arr[7] & arr[2] arr[7] = -10 $\Rightarrow$ arr[-11]; <i>Out of bounds / Invalid data / goto next</i>
8	arr[8] = 9 # goto next	
9	arr[9] = 12	arr[9] = 12 $\Rightarrow$ arr[11] <i>Out of bounds / Invalid data / goto next</i>
10	# stop	

After Modification:

arr[] = {

0	1	2	3	4	5	6	7	8	9
1	2	3	3	5	6	7	-10	9	12
			-14						

arr = 4.

```
int firstMissing(vector<int> &ar) {
```

```
    int N = ar.size();
```

```
    for (int i = 0; i < N; i++) {
```

```
        # Bring correct data to index i.
```

```
        while (ar[i] != i+1) {
```

```
            int ele = ar[i];
```

```
            int ind = ele-1;
```

```
            if (ind < 0 || ind == N) { break; } → #inherent data
```

```
            if (ar[i] == ar[ind]) { break; } → #duplicate
```

```
            swap ar[i] & ar[ind];
```

```
        }
```

```
    }
```

```
    for (int i = 0; i < N; i++) {
```

```
        if (ar[i] != i+1) {
```

```
            return i+1;
```

```
        }
```

```
    return N+1;
```

```
}
```

Edge Case:

ar[5] = { ~~0~~ 1 ~~2~~ 3 4 }  
~~3~~ 3  
 3

i Correct Data Take data to correct position.

0 ar[0] = 4 ar[0] = 4 ⇒ ar[3] swap ar[0] & ar[3]

ar[0] = 3 ar[0] = 3 ⇒ ar[2] swap ar[0] & ar[2]

ar[0] = 3 ar[0] = 3 ⇒ ar[2] swap ar[0] & ar[2]

# Note: if elements to swap are same we break.

int firstMissing(vector<int> &arr) {

int N = arr.size();

for (int i = 0; i < N; i++) {

# Bring current data to index i.

while (arr[i] != i+1 || arr[i] >= 1 || arr[i] <= N || arr[i] != arr[arr[i]-1])

swap arr[i] & arr[arr[i]-1] # index

3

3

Total Iterations =  $N + N = 2N$

Total Outerloop =  $\{0, N-1\} = N$

+

Total Innerloop =  $N$

a. 1 innerloop itera = 1 swap

b. 1 swap will bring atleast 1 element to its correct position

c. At most we have  $N$  swaps

# Total Inner loop Iterations =  $N$  Swaps

for (int i = 0; i < N; i++) {

if (arr[i] != i+1) {

return i+1;

3

return N+1;

3

#Note: If there is break in 2nd loop

calculate iterations & estimate

Big O