

Konstruktorok tesztelése	Paraméter nélküli konstruktor: _h=0, _n=100 Paraméteres konstruktor: m=3 -> _h=0, _n=3 m=0, m=-1 -> HIBAS_MERET
üres_e()	Konstruktorok után közvetlenül meghívva igazat ad: PrSor Q1; _h=0, _n=100, _vec=[] -> true PrSor Q2(3); _h=0, _n=3, _vec=[] -> true Betteszünk egy elemet Q1-be, hamisat ad: Elem e(1,"a"); Q1.prSorba(e); _h=1, _n=100, _vec=[<1,"a">] -> false Kivesszük az előbb berakott elemet, újra igazat ad: Q1.maxKivesz(); _h=0, _n=100, _vec=[] -> true
prSorba()	Létrehozunk egy 2 hosszú sort, majd 5 elemet berakunk: PrSor Q(2); Elem e1(1,"a"), Elem e2(2,"b"), Elem e3(3,"c"), Elem e4(4,"d"), Elem e5(5,"e") Q.prSorba(e1); _h=1, _n=2, _vec=[ <1,"a">] Q.prSorba(e2); _h=2, _n=2, _vec=[<1,"a">, <2,"b">] Q.prSorba(e3); _h=3, _n=4, _vec=[<1,"a">, <2,"b">, <3,"c">] Q.prSorba(e4); _h=4, _n=4, _vec=[<1,"a">, <2,"b">, <3,"c">, <4,"d">] Q.prSorba(e5); _h=5, _n=8, _vec=[<1,"a">, <2,"b">, <3,"c">, <4,"d">, <5,"e">] Méret megadás nélkül létrehozunk egy prioritásos sort, és berakunk 2 elemet: PrSor Q1; Q.prSorba(e1); _h=1, _n=100, _vec=[ <1,"a">] Q.prSorba(e2); _h=2, _n=100, _vec=[<1,"a">, <2,"b">]

maxindex()	<p>Ezen alapszik a max() és maxKivesz() metódusok helyes működése, így ennek tesztelésével érdemes elsőként foglalkozni. Mivel ezt a függvényt csak h&gt;0 esetben hívja meg az osztály, nem kell az üres intervallum esettel foglalkozni.</p> <p>Elemszám szerint: egy elemű (n most tetszőleges, egy *-gal jelöljük):  _h=1, _n=*, _vec=[ &lt;1,"a"&gt;] -&gt; ind=0</p> <p>Több elemű esetek:</p> <p>első a legnagyobb:  _h=3, _n=*, _vec=[&lt;5,"a"&gt;, &lt;2,"b"&gt;,&lt;3,"c"&gt;] -&gt; ind=0</p> <p>utolsó a legnagyobb:  _h=3, _n=*, _vec=[&lt;1,"a"&gt;, &lt;2,"b"&gt;,&lt;3,"c"&gt;] -&gt; ind=2</p> <p>belső a legnagyobb:  _h=3, _n=*, _vec=[&lt;1,"a"&gt;, &lt;3,"c"&gt;,&lt;2,"b"&gt;] -&gt; ind=1</p> <p>nem egyértelmű:  _h=3, _n=*, _vec=[&lt;5,"a"&gt;, &lt;2,"b"&gt;,&lt;5,"c"&gt;] -&gt; ind=0  _h=3, _n=*, _vec=[&lt;1,"a"&gt;, &lt;3,"b"&gt;,&lt;3,"c"&gt;] -&gt; ind=1  _h=3, _n=*, _vec=[&lt;3,"a"&gt;, &lt;3,"b"&gt;,&lt;3,"c"&gt;] -&gt; ind=0</p>
maxKivesz()	<p>Üres sor esete:  _h=0, _n=*, _vec=[] -&gt; URES_PRSOR</p> <p>Egy-elemű :  _h=1 _n=*, _vec=[ &lt;1,"a"&gt;] -&gt; _h=0 _n=*, e=&lt;1,"a"&gt;, (még ezt is tudjuk, de ilyenkor a tömb már „üres”: _vec=[&lt;1,"a"&gt;])</p> <p>Több elemű esetek:</p> <p>első a legnagyobb:  _h=3, _n=*, _vec=[&lt;5,"a"&gt;, &lt;2,"b"&gt;,&lt;3,"c"&gt;] -&gt;  e=&lt;5,"a"&gt;, _h=2, _n=*, _vec=[&lt;3,"c"&gt;, &lt;2,"b"&gt;]</p> <p>utolsó a legnagyobb:  _h=3, _n=*, _vec=[&lt;1,"a"&gt;, &lt;2,"b"&gt;,&lt;3,"c"&gt;] -&gt;  e=&lt;3,"c"&gt;, _h=2, _n=*, _vec=[&lt;1,"a"&gt;, &lt;2,"b"&gt;]</p> <p>belső a legnagyobb:  _h=3, _n=*, _vec=[&lt;1,"a"&gt;, &lt;3,"c"&gt;,&lt;2,"b"&gt;] -&gt;  e=&lt;3,"c"&gt;, _h=2, _n=*, _vec=[&lt;1,"a"&gt;, &lt;2,"b"&gt;]</p> <p>nem egyértelmű:  _h=3, _n=*, _vec=[&lt;5,"a"&gt;, &lt;2,"b"&gt;,&lt;5,"c"&gt;] -&gt;  e=&lt;5,"a"&gt;, _h=2, _n=*, _vec=[&lt;5,"c"&gt;, &lt;2,"b"&gt;]  _h=3, _n=*, _vec=[&lt;1,"a"&gt;, &lt;3,"b"&gt;,&lt;3,"c"&gt;] -&gt;  e=&lt;3,"b"&gt;, _h=2, _n=*, _vec=[&lt;1,"a"&gt;, &lt;3,"c"&gt;]  _h=3, _n=*, _vec=[&lt;3,"a"&gt;, &lt;3,"b"&gt;,&lt;3,"c"&gt;] -&gt;  e=&lt;5,"a"&gt;, _h=2, _n=*, _vec=[&lt;3,"c"&gt;, &lt;3,"b"&gt;]</p>

max()	Hasonló maxKivesz()-hez. Gyakorlaton nem néztük (ennek ellenére beadandóba fokozottan ügyeljünk, hogy minden metódussal kell tesztelést végezni).
PrSor(PrSor)	<p>A copy konstruktor és az értékadó operátor tesztelése hasonló. Ennek mintájára készíthetjük el az értékadó operátorét is, hozzávéve azt az esetet, amikor a PrSornak az önmagával vett értékadását vizsgáljuk.</p> <p>Például: létrehozunk egy prioritásos sort, beleteszünk mondjuk két elemet. Ezt lemásolva létrehozunk egy új prioritásos sort, majd mindkettőbe beteszünk, kiveszünk. Például:</p> <p>Elem e1(1,"a"), Elem e2(2,"b"), Elem e3(3,"c"), Elem e4(4,"d");</p> <p>PrSor Q(5);</p> <p>Q.prSorba(e1); Q.prSorba(e3);</p> <p>PrSor QM(Q);</p> <p>QM sor változói: _h=2, _n=5, _vec=[&lt;1,"a"&gt;,&lt;3,"c"&gt;]</p> <p>Q.prSorba(e2) -&gt; _h=3, _n=5, _vec=[&lt;1,"a"&gt;,&lt;3,"c"&gt;,&lt;2,"b"&gt;]</p> <p>QM.prSorba(e4) -&gt; _h=3, _n=5, _vec=[&lt;1,"a"&gt;,&lt;3,"c"&gt;,&lt;4,"d"&gt;]</p> <p>e=Q.maxKivesz() -&gt; e=&lt;3,"c"&gt;, _h=2, _n=5, _vec=[&lt;1,"a"&gt;,&lt;2,"b"&gt;]</p> <p>e=QM.maxKivesz() -&gt; e=&lt;4,"d"&gt;, _h=2, _n=5, _vec=[&lt;1,"a"&gt;,&lt;3,"c"&gt;]</p> <p>Üres sorról készült másolat</p> <p>PrSor Q1;</p> <p>PrSor Q1M(Q1);</p> <p>Q1M sor változói: _h=0, _n=100</p>