

```
mtest2 := MC[Pp[3, 1], Pp[2, 7], Pp[10, 4], Pp[9, 5], Pp[6, 8], Pp[11, 7], Pp[10, 9]];
Crossings[mtest2]
```

```
11
```

```
{IsCrossing[mtest2, 10, 4], IsCrossing[mtest2, 4, 10], IsCrossing[mtest2, 10, 28]}
{True, True, False}
```

```
IsCrossing[mtest2, #[[1]], #[[2]]] & /@ {{10, 4}, {4, 10}, {10, 5}}
{True, True, False}
```

```
IsCrossing[mc_MC, p1_, p2_] := MemberQ[mc, (Pp | Pm | Po)[p1, p2] | _[p2, p1]]
```

```
CrossingDirection[mtest2, #[[1]], #[[2]]] & /@ {{10, 4}, {4, 10}, {10, 5}}
{1, -1, 0}
```

```
CrossingDirection[mc_MC, p1_, p2_] := Which[
  IsDirectedCrossing[mc, p1, p2], 1,
  IsDirectedCrossing[mc, p2, p1], -1,
  True, 0]
```

```
CrossingPosition[mtest2, #[[1]], #[[2]]] & /@ {{10, 4}, {4, 10}, {10, 5}}
{0, 0, 0}
```

```
CrossingPosition[mtest2, 2, 7]
```

```
2
```

```
CrossingPosition[mc_MC, p1_, p2_] :=
  Position[mc, (Pp | Pm | Po)[p1, p2] | _[p2, p1]][[1, 1]]
```

```
W1[mtest2, 2, 7]
```

```
3
```

```
mc = mtest2; p1 = 2; p2 = 7;
```

```
Total[
  Cases[mc, (P_) [i_, j_] /; i > p2 && p1 < j ≤ p2 ⇒ P] /. {Pp → 1, Pm → -1, Po → 0}
]
{1, 1, 1}
```

```
Total[{1, 2, 3}]
```

```
6
```

```
Plus@@{1, 2, 3}
```

```
6
```

A better data structure:

```
mtest6 = MC[R3[11, 2, 7], Xp[3, 1], Xm[4, 5]];
Res1[MC[R3[i_, j_, k_], rest___]] :=
  MC[Xp[i, j], Xp[i + 0.3, k], Xm[j + 0.3, k - 0.3], rest]

Renormalize[mc_MC] := Module[
  {labels = Union[Cases[mc, _, {2}]]},
  mc /. Thread[labels → Range[Length[labels]]]
];
Res1[MC[R3[i_, j_, k_], rest___]] := Renormalize[
  MC[Xp[i, j], Xp[i + 0.3, k], Xm[j + 0.3, k - 0.3], rest]
]

Res1[mtest6]
MC[Xp[9, 2], Xp[10, 8], Xm[3, 7], Xp[4, 1], Xm[5, 6]]
```